

A PUMA 600 ROBOT  
USED AS A JOYSTICK CONTROLLER FOR TELEOPERATIONS--  
AN EXPERIMENTAL STUDY

BY

CHING-SHYONG SHIEH

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL  
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT  
OF THE REQUIREMENTS FOR THE DEGREE OF  
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1990

#### ACKNOWLEDGEMENTS

The author wishes to manifest his gratitude to his committee chairman, Dr. Gary Matthew, for his encouragement and guidance throughout this work. Special thanks are given to Dr. Joseph Duffy and Dr. Carl Crane for their support and patience. The author would also like to thank Dr. George Sandor and Dr. Keith Doty as members of the supervisory committee. The author is also grateful for Dr. Scott Smith's attendance in the dissertation defense.

Furthermore, the author would like to express his appreciation to CIMAR's facilities and experience in manual controllers.

## TABLE OF CONTENTS

	<u>page</u>
ACKNOWLEDGEMENTS.....	ii
TABLE OF CONTENTS.....	iii
ABSTRACT.....	iv
CHAPTERS	
1 INTRODUCTION.....	1
1.1 Review of Teleoperator Systems.....	1
1.2 Classification of Teleoperator Systems.....	5
1.3 The Robot as a Teleoperator Controller.....	12
2 THE PUMA 600 ROBOT.....	15
2.1 System Characteristics.....	16
2.2 Forward Displacement Analysis.....	22
3 SYSTEM DEVELOPMENT AND CONCEPTUAL DESIGN.....	30
3.1 Review of Force-Sensed Control of Robots.....	30
3.2 Joystick Subsystem.....	35
3.3 Application of Jacobian Matrix.....	42
3.4 Stability Analysis.....	50
4 EXPERIMENTAL IMPLEMENTATION AND RESULTS.....	64
4.1 On-Desk Sensor.....	65
4.2 Experimental of Single-Axis System.....	70
4.3 Experimental Observations for the PRJS.....	82
4.4 Simulation of Remote Force Reflection Joystick..	84
4.5 Milestones for Achieving Experimental Results...	87
5 CONCLUSIONS AND RECOMMENDATIONS.....	92
5.1 Concluding Remarks.....	92
5.2 Recommendations and Future Work.....	94
REFERENCES.....	97
BIOGRAPHICAL SKETCH.....	102

Abstract of Dissertation Presented to the Graduate School  
of the University of Florida in Partial Fulfillment of the  
Requirements for the Degree of Doctor of Philosophy

A PUMA 600 ROBOT  
USED AS A JOYSTICK CONTROLLER FOR TELEOPERATIONS--  
AN EXPERIMENTAL STUDY

BY

Ching-Shyong Shieh

December 1990

Chairman: Dr. Gary K. Matthew

Major Department: Mechanical Engineering

This project involves a new approach to teleoperations, utilizing an industrial robot as a joystick controller. The concept is developed based on the fact that robot systems and force-reflecting joystick controllers essentially share the same technology foundation. This concept has been initiated with a PUMA 600 robot to demonstrate its feasibility.

In this experimental system, a commercially available six degree-of-freedom force/torque sensor is installed at the wrist of the PUMA 600 robot to enhance the factory-supplied system with an external control loop in order to accommodate the operator's application of forces and torques. The Jacobian matrix associated with the configuration of the robot

is used to transform the measured wrenches into joint torques. Several trial strategies, which use the resulting joint torques as different order functions of joint displacements, are tested to convert these joint torques into commands so that the robotic joystick can be moved accordingly.

Experimental results showed that, with the current facilities, the PUMA 600 robot has the capability to be used as a joystick. An experiment which reflects artificial forces from the remote unit to the robotic joystick subsystem is also conducted. This remotely reflecting force can be well sensed by the operator. Some observations in the experiments are discussed so that a system with better performance can be achieved.

## CHAPTER 1 INTRODUCTION

### 1.1 Review of Teleoperator Systems

Teleoperation indicates direct human control of a remote manipulator. The objective of teleoperation is to carry out remotely complicated non-repeated tasks in circumstances where preprogramming is severely limited.

Teleoperation began with the handling of radioactive materials in nuclear hotcells in the late 1940s. It was not until 1966 that the term "teleoperator system" was introduced for the first time by Johnsen [1]. Teleoperator systems were then brought to the areas of submersibles and space exploration.

Roy Goertz and his colleagues at the Argonne National Laboratory (ANL) developed the first mechanical unilateral master-slave teleoperator system for handling radioactive materials in 1947 [2]. A year later, the first bilateral system was developed based on mechanical coupling. This system showed that force<sup>1</sup> feedback was critical if the operator was required to perform precise tasks. However, the performance of the mechanically linked system was found to be

---

<sup>1</sup> Throughout this work, the term force or alternately wrench has a general meaning of "force and torque."

limited by the characteristics of the transmission, the increasingly difficult requirements in the working environments, and the distance between the controller and the remote site. In 1954, an electrical bilateral master-slave manipulator system was constructed at ANL which provided position and force feedback. Surprisingly, since the inception of this electrical bilateral system, no other teleoperator system tested has been able to perform more effectively. Although the bilateral master-slave system represents the demonstrated state-of-the-art, its performance is still 2 to 15 times slower than that of humans, dependent primarily on the task complexity [3].

As developments proceeded, flexibility of universal joystick controllers, or indeed, computer-aided manual controllers, became a desired feature. The concept behind universal joystick controllers is that the controllers can be better designed to interface with the human operator, while remote manipulators can be optimized to meet their functional needs. A universal controller is able to locate a remote manipulator of the same or fewer degrees of freedom in space. If additional degrees of freedom exist in the remote manipulator, there must, of course, be added algorithms to deal with the redundant degrees of freedom. There need be no geometrical resemblance between the master controller and the remote manipulator. Kinematic transformations, performed by computers, are required to coordinate the motions between the

control device (joystick handgrip) and the remote manipulator (tool).

In the Center for Intelligent Machines and Robotics (CIMAR) at the University of Florida, a nine-string universal joystick controller and its computational base have been designed and fabricated [4, 5]. This unilateral joystick system is interfaced to a six-degree-of-freedom (6-DOF) MBA prototype manipulator. The joystick consists of a handgrip suspended by nine strings connected to spring-loaded potentiometers and a computer interface. The potentiometers track the lengths of the unwound strings and use these lengths to calculate the location and orientation of the handgrip. Force reflection is not available with this device. Real-time computer control capabilities have also been implemented on a PDP-11/03 computer and a PDP-11/23 computer connected by a shared-memory architecture. By simply changing the software package, one is able to use the controller to drive a manipulator with completely different geometry. If the strings tangle, the computed location and orientation will be incorrect. When released, the handgrip automatically returns to approximately the center of the joystick frame, due to the loads of the springs.

A system similar to the nine-string universal joystick controller is under development at the University of Texas, Austin. It adds a force-reflecting capability. Three gimbaled pneumatic cylinders are installed to apply constant

forces to the handgrip and nine strings are connected to DC-motors with transducers for force control and position measurement [6].

Although the control device and the remote manipulator normally have the same degrees of freedom, this is not a requirement. A planar 4-DOF force-reflecting controller was designed and constructed at CIMAR to control the horizontal planar motion of the 6-DOF MBA manipulator [7]. The joystick consisted of four links serially connected by revolute joints. A cascading arrangement of chains and sprockets connected each joint to a base-mounted assembly of an actuator, a potentiometer, and torque sensors. Since planar motion requires only three degrees of freedom, the redundant degree of freedom was used either to optimize some criteria such as the actuator torques or to configure the controller so that mathematical singularities were avoided [8, 9]. Substantial friction and backlash exist in this controller.

Another 6-DOF universal position joystick with the shape of scissors was designed and fabricated in association with CIMAR. This joystick has a balanced feature [10]. It was used to drive the MBA manipulator with a real-time computer implementation.

A well-known example of a 6-DOF force-reflecting universal controller is the one used at the Jet Propulsion Laboratory (JPL) [11]. This device, although not the first attempt, represents the first fully developed 6-DOF force-

reflecting universal joystick system. The joystick is based on a spherical geometry. It features approximately a one-cubic-foot workspace and has a counterbalanced structure.

Hirzinger discussed about how a force transducer ball could be used to drive a robot for teaching [12, 13, 14, 15, 16]. The transducer ball could be attached to the gripper working with another force/torque sensor and thereby allow the operator to move the robot. It could also be mounted on a desk and used as an isometric rate joystick. Subjective/objective evaluations of the device were not presented in any of the above references.

### 1.2 Classification of Teleoperator Systems

A teleoperator system consists of three principal elements: the control unit, the remote unit, and the communication channel. The remote unit generally consists of a mechanical arm with an end-effector to grasp objects. The control unit must contain, at a minimum, a device for the operator to input instructions for the desired activities of the remote unit. The communication channel allows required information to be passed between the control unit and the remote unit in forms compatible to both of them.

The human operator in the teleoperator system provides the highest level of decision-making capability. The human operator must be aware of the ongoing task of the remote manipulator and be able to command the remote manipulator

effectively. Because of the completely different natures between the human and the manipulator, an effective control mediator is needed to augment the human operator. This mediator is called the man-machine interface and consists of the control unit and part of the communication channel. Intelligent sensing capabilities can be integrated into the man-machine interface to enhance its performance.

The types of man-machine interfaces used in teleoperator systems can be roughly categorized as follows, although combinations of them do occur:

- 1) Simple function controllers are usually compact, cheap, and thereby work with limited functions. They are only used to control a single joint or a single degree of freedom. Switches and potentiometers are typical examples of this type of controller. Switch controllers generally consist of a spring-centered, three-position, discrete action device. Potentiometers are commonly used to adjust signal gains. To control a manipulator with combinations of single degree-of-freedom movements, the difficulty varies with the joint arrangement and the desired path of motion; for example, a prismatic joint is easily moved in a straight line motion along its axis. On the other hand, an arbitrary straight path is not easily generated by a revolute manipulator.

- 2) Master-slave controllers are either a geometrical duplicate or a scaled replica of the remote manipulator. The one-to-one correspondence of this type of controller with its slave manipulator arguably reduces the operator's workload. It also allows the system to be implemented without using a computation base. Force reflection from the remote manipulator is sometimes incorporated in the master-slave system, which enhances the interface transparency. Commercially available bilateral joysticks are often master-slave controllers.
- 3) Anthropomorphic controllers are very much like other types of master-slave controllers, except that it has the configuration of the human arm. It utilizes natural human motion to control the remote manipulator, which results in a reduced learning period of time for the operator. When properly designed, it can control as many as seven independent degrees of freedom: 3 for the shoulder, 1 for the elbow, and 3 for the wrist. Of course, this type of controller is restricted by the physical dimensions of the human arm.
- 4) Universal joystick controllers are able to control the remote manipulator even when there is no geometrical analogy between them. A computation base is mandatory in order to interface the controller with a manipulator arm. Force reflection from the remote manipulator can be incorporated. Some universal joysticks are classified

according to their particular features. An isotonic joystick is a position-operated, fixed-force (isotonic) device used to control two or more degrees of freedom with a single hand within a limited operational volume. An isometric joystick is a force-operated, minimal-displacement (isometric) device used to control two or more degrees of freedom with a single hand. The output of the controller corresponds directly to the forces applied by the operator and drops to zero unless manual force is maintained. A proportional joystick is a single-handed, two or more degree-of-freedom device with a limited operational volume in which the displacement is in proportion to the force applied by the operator.

As far as the flow of information is concerned, teleoperator systems can also be distinguished as unilateral or bilateral systems. In unilateral systems, position commands are fed from the master to the slave, with no reverse information flow through the communication channel. The operator is obliged to use visual sense as well as experience to assess the results of his commands. In bilateral systems, in addition to the position information being sent from the master to the slave, applied loads on the slave are reflected to the master device. Bejczy and Salisbury found that task completion times were often reduced by 40% when the operator was given force feedback information [11].

Based on the control strategy, teleoperator systems can be classified as direct rate control, resolved motion rate control, direct position control, and resolved motion position control.

Direct rate control represents an elementary type of controller which controls the velocities of actuators of the manipulator with a one-to-one correspondence. No computational interface is required since rate controllers do not address the coordinated motion of the manipulator end-effector. Force feedback cannot be accommodated. Hence, they are much less desirable as a transparent interface device.

The next advancement in teleoperator systems is resolved motion rate control, first described by Whitney during the late 1960s [17]. Differing from direct rate control where the operator controls the velocities of particular joints of the manipulator, resolved motion rate control calculates the joint velocities of the actuators so that the end-effector moves along a trajectory at a given velocity with respect to the task environment. Jacobian transformation is a commonly used technique. Force feedback capability is limited or nonexistent in resolved motion rate control systems.

A common weakness with all the rate controllers is that positioning of the end-effector in space is indirect due to the velocity control strategy. The operator must mentally integrate the velocity and try to command the manipulator to reach a desired position. In general, rate controllers cannot

effectively command the coordinated motion of a slave device in space. Rate control is 10 to 50 times slower than the master-slave position control for some tasks [3].

Direct position control is usually used by a replica device of the remote manipulator. Resolved motion position control is a strategy used by the universal controller.

A comparison between the performance of a rate controller and that of a replica master controller with force feedback has been documented by Wilt et al. [18]. Different control modes were implemented using the same remote manipulator but with the two different controllers. The results of task completion time with tasks of varying difficulty showed that bilateral control had a distinct advantage over resolved motion rate control.

Brooks concludes that position control is the best for fine motions where accuracy is of paramount importance, while rate control is the best for large gross transfer motion [19]. Rate control has the disadvantage of loss of coordinate reference when the operator attempts to command an angular velocity with non-zero components along more than one of the reference axes.

A more recently developed term is supervisory control which, in general, allows the operator to specify some of the desired tasks symbolically. The computer interprets the issued symbolic commands to instruct a remote device. This

type of man-machine interface results in a hybrid between a teleoperator system and an autonomous robot.

The communication channels between the control unit and the remote unit can also be classified into three types:

- 1) pure mechanical transmission,
- 2) electrical transmission, and
- 3) computer-aided teleoperation.

Mechanical teleoperator systems are the earliest development and are still used in the majority of current nuclear installations. The transmissions can be tapes, shafts, cables, etc. Friction, backlash and elastic deflections are undesirable features of these systems. Master controllers can not be far away from slave manipulators with mechanical transmissions. Force feedback can be incorporated, however.

Electrical teleoperator systems allow greater flexibility in system layout and better versatility in control. One-to-one correspondence is kept between the master device and the remote manipulator. Both unilateral and bilateral systems exist.

When computer-aided teleoperation is used as the communication channel, the one-to-one correspondence between the master device and the remote manipulator is no longer significant and is replaced by transformations between the

two operational spaces. The system can be made more flexible at the cost of computational delay and data communication delay. It is realized that the time delay in a computer-aided teleoperator system is critical to system stability.

As of today, neither a communication architecture nor a control strategy exists which can eliminate the time delay. The direction of research in remote manipulators is approaching modular designs so the components can easily be replaced at a remote site [20]. As far as the control unit is concerned, there are many different models, and several authors [3, 21, 22] give extensive reviews about the evolution of teleoperator systems.

### 1.3 The Robot as a Teleoperator Controller

Joystick technology, being used as a man-machine interface in teleoperation, attempts to utilize machine intelligence to augment the operator and thus reduce his efforts during operation. Since the human being and the manipulator are by nature quite different, it is the function of the joystick to transfer necessary information between these two entities in a way suitable to both. The common approach to joystick design is to follow some design guidelines which are obtained from experience and to make the joystick meet the performance requirements. Chances are that some conflicts among different performance requirements arise, and thus compromises must be made in the design process.

The philosophy introduced here is that, instead of designing a whole new edition of a joystick, a robot system is used in place of a joystick controller, or even a force-reflecting joystick. Based on this idea, the technologies of robot design and joystick design are merged. In the field of joystick research, it is also possible to set up a testbed for different types of joysticks as long as the same types of robot systems exist. These testing results can be helpful for those people who design joysticks with selected criteria. With the robotic joystick system, some of the important requirements for joystick design are internally taken care of by the original robot system.

This research specifically intends to utilize an industrial robot, a PUMA 600, as a controller device for teleoperations. A wrist force/torque sensor is installed to sense the operator's application of wrenches. These sensed wrenches are further transformed into joint torques via Jacobian transpose mapping in real time. A suitable control algorithm is selected amongst several choices so that the PUMA robot is able to perform effectively as a joystick controller. Moreover, an experiment using the PUMA-robot joystick subsystem (PRJS) as a bilateral system is conducted. A remote computer, which simulates the reflecting joint torques, is integrated with the PRJS to demonstrate its capability as a bilateral device. In subsequent chapters, we detail the development of the PRJS and an experimental implementation.

Chapter 2 gives the necessary background of the characteristics of the PUMA 600 robot system so that it can be used as a joystick after some modifications. Also discussed is a primitive approach to the forward displacement analysis of the PUMA 600 robot, which fits the factory-supplied measurement system.

Chapter 3 introduces the PRJS. A brief review of force control in robotics is conducted. The Jacobian matrix and the transformations which integrate the sensed wrenches into the system are conducted based on a practical implementation of screw theory. System dynamics are analyzed to indicate potentially acceptable systems.

Chapter 4 explores different algorithms for the PRJS. The performance results of stable algorithms are further analyzed. To investigate force-reflection on the device, artificial force information is brought into the PRJS from a remote computer, which demonstrates the potential for using an industrial robot as a force-reflecting joystick.

Chapter 5 concludes this project with a comparison of advantages and disadvantages of a robotic joystick subsystem. Some ideas for future development of robotic joystick subsystem are also discussed.

## CHAPTER 2 THE PUMA 600 ROBOT

Industrial robots are designed to satisfy kinematic specifications. The inertial parameters, such as the location of center of mass and the moment of inertia of each link, are incidentally attributed and are usually not known even to the manufacturers of the robots. Industrial robots are also characterized by high gear ratios and substantial friction. As a result, most industrial robots are designed with simple approaches by utilizing independent-joint position-integral-derivative (PID) controllers with the sacrifice of faster speed and better payload handling capability.

In the CIMAR laboratory, a PUMA 600 robot is available for this project. It offers six degrees of freedom which can span the translational and rotational coordinates in 3-space. Although position scaling and re-referencing allow the operational volume of a joystick to be sized to the reach of a human so that the joystick can drive a slave manipulator throughout its full workspace, the concept of synchronized operation prefers the dexterity of the joystick to be as great as that of the controlled manipulator.

In order for an industrial robot to serve as a research tool, it is necessary to modify the control system so that the

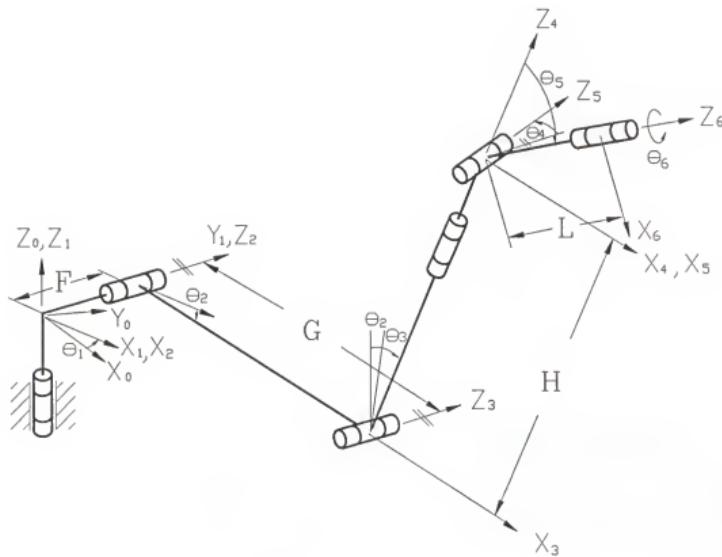
robot can be driven by a computer using a standard computer language. Different methods can make this possible. In this research, the existing servo of the PUMA robot is adopted. Most of the originally design features of the PUMA system are preserved, except for the DEC-11/02 CPU and some components which are connected directly to the CPU [23].

In this chapter, the factory-supplied PUMA 600 robot system is briefly described. Communication between the PUMA controller and a personal computer is briefly explained. Since the PUMA robot is to be used as a joystick device, location and orientation of its end-effector must be calculated via a forward displacement analysis.

### 2.1 System Characteristics

The PUMA robot is designed to be a Programmable Universal Machine for Assembly. The PUMA 600 robot has six serially connected revolute joints and a wrist-mounted gripper. It can handle a payload only up to five pounds. The kinematic parameters of the PUMA 600 robot are shown in Figure 2.1.

The PUMA 600 robot is an independent-joint position-controlled device with pre-specified PID control gains at the hardware level. It is not normally backdrivable. The actuator of each joint incorporates a pair of position sensors, a potentiometer and a 16-bit optical incremental encoder. The coarse-resolution potentiometers are used to calibrate the power-up joint positions of the robot. The



Legend:

$$F = 5.88" \quad G = 17" \quad H = 17.05" \quad L = 6"$$

Figure 2.1 Schematic Diagram of PUMA 600 Robot

fine-resolution incremental encoders are used with the servos to accurately control the joints. Accuracy and repeatability of the PUMA 600 robot are both claimed by the manufacturer to be within 0.1 mm [24]. The reason for using two position measurement systems is simply because the price of potentiometers and incremental encoders is less expensive than that of absolute encoders with the same resolution.

Since potentiometers are absolute devices, when power is turned on and a voltage is applied to them, output voltages can be measured from their wipers to indicate joint angles of the robot. Potentiometers are connected to the 8-bit analog-to-digital converters providing approximate joint angles (to the order of 2 to 3 degrees). The 16-bit incremental encoders measure joint angles more precisely (to the order of 0.05 degrees). Whenever power is shut down, the robot system loses track of the measurement of joint angles due to the use of the incremental encoders. It is necessary to settle the measurement references of the encoders with the help of potentiometers so that the PUMA robot is precisely controlled. Initialization and calibration routines for the PUMA 600 robot are described in Shieh [23] along with the system parameters used by the VAL operating system.

The PUMA system was designed in the early 1970s. As a result, it is sometimes not powerful or flexible enough for today's demands. In particular, it supplies only very limited communication with the outside world. The PUMA system, either

with VAL or its supplementary VAL-II, offers only RS-232C serial communication lines with the transmission rate not higher than 9600 baud. The situation is made worse by the fact that the signals passed through the serial line must be synchronized with the control cycle and be compatible with the formats of the VAL system, which is not always economical.

The components in the PUMA 600 controller are made up of Unimation components which include an arm interface board (AIB), 6 servo boards, and 6 power amplifiers, and Digital Equipment Corporation (DEC) components which include an LSI-11/02 CPU, 2K EPROM, 4K RAM, a serial interface to a terminal, and a DRV-11 parallel interface to the AIB. The basic layout of the controller is shown in Figure 2.2. The EPROM is the place where VAL codes and constants are stored. The factory-supplied PUMA system needs to be taught by a teach pendant working together with the VAL software. There is no way to incorporate force information into the system.

In order to communicate with the servo of the PUMA robot, the communication protocols used by the VAL system must be emulated. The communication protocols are given in Table 2.1, which have been verified with the help of a logic analyzer and the on-line debug tool (ODT) of the PDP computer.

The joint microprocessor commands of the VAL system are encoded in the format of Table 2.2, where bits 0-2 are used for joint selection and bits 3-6 are used for different command modes. The command  $00X_8$  represents the position mode

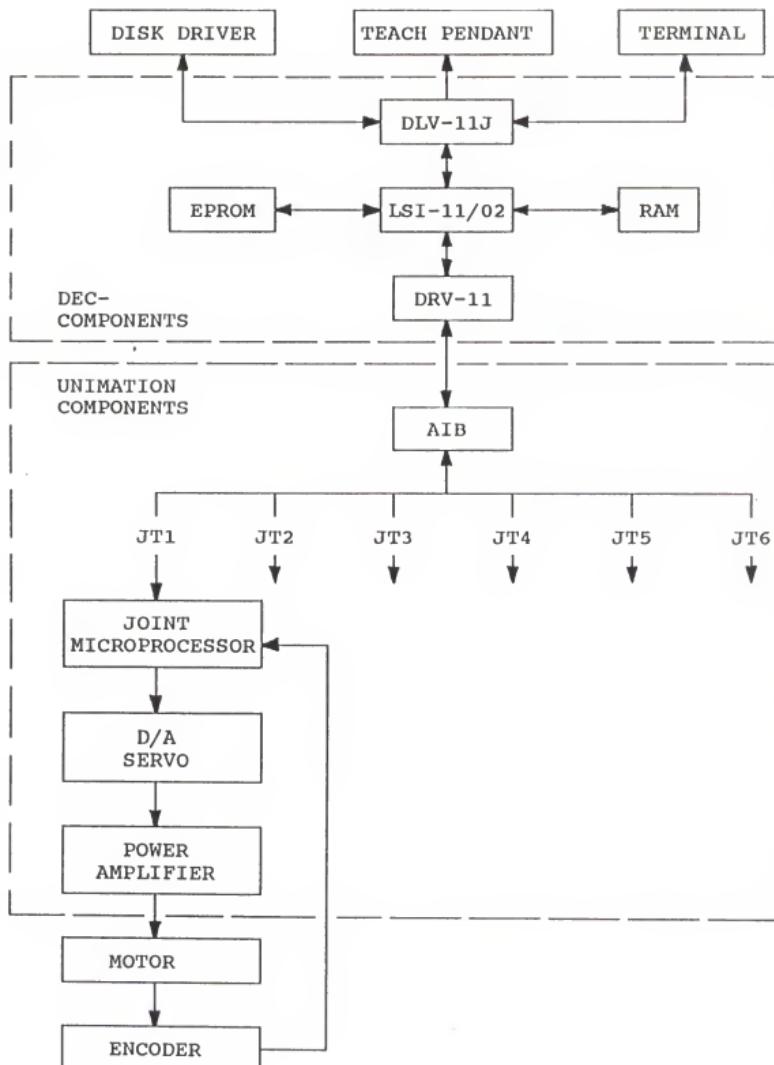


Figure 2.2 Layout of the PUMA 600 Controller

Table 2.1 Communication Protocols with PUMA Joint Microprocessors

WRITE:

Operations	Functions
CSR0 $\leftarrow \equiv 0$	Clear CSR0 bit
OUTREG $\leftarrow \equiv$ Command	Send out the command
CSR0 $\leftarrow \equiv 1$	Set CSR0 bit
OUTREG $\leftarrow \equiv$ Data	Write data out

READ:

Operations	Functions
CSR0 $\leftarrow \equiv 0$	Clear CSR0 bit
OUTBUF $\leftarrow \equiv$ Command	Send out the command
CSR0 $\leftarrow \equiv 1$	Set CSR0 bit
Data $\leftarrow \equiv$ INREG	Read data in

Table 2.2 Bit-Formats of Microprocessor Commands of PUMA robot

15	Not Used		8	7	6	3	2	0													
Sequential bit																					
Command mode bits																					
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">0000</td> <td style="width: 50%;">Position mode</td> </tr> <tr> <td>1100</td> <td>Read encoder</td> </tr> </table>		0000	Position mode	1100	Read encoder																
0000	Position mode																				
1100	Read encoder																				
Joint selection bits																					
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">000</td> <td style="width: 50%;">Command joint 1</td> </tr> <tr> <td>001</td> <td>Command joint 2</td> </tr> <tr> <td>010</td> <td>Command joint 3</td> </tr> <tr> <td>011</td> <td>Command joint 4</td> </tr> <tr> <td>100</td> <td>Command joint 5</td> </tr> <tr> <td>101</td> <td>Command joint 6</td> </tr> </table>		000	Command joint 1	001	Command joint 2	010	Command joint 3	011	Command joint 4	100	Command joint 5	101	Command joint 6								
000	Command joint 1																				
001	Command joint 2																				
010	Command joint 3																				
011	Command joint 4																				
100	Command joint 5																				
101	Command joint 6																				

which can drive joint X-1 to an assigned joint position. The command  $14X_8$  is used to read the encoder code of joint X-1. The subscript following the number denotes the number base system; for example, "<sub>8</sub>" indicates an octal number. Encoder codes can be further converted into joint angles.

As mentioned, the PUMA robot is essentially controlled with an "independent-joint" strategy. The VAL commands, after interpretation by the operating system, are sent to joint microprocessors through a DRV-11 parallel interface and an AIB every 28 msec. The correspondingly commanded joint positions are then interpolated into 32 equal increments to servo the joint actuators every 0.875 msec. It is very important for the control codes to complete execution within the 28 msec cycle time so that the robot can move smoothly.

## 2.2 Forward Displacement Analysis

With a universal joystick, the geometric relationship between the joystick and the slave manipulator does not normally keep a one-to-one correspondence. Therefore, a forward displacement analysis of the joystick and a reverse displacement analysis of the slave manipulator are required in order to coordinate these two subsystems. To use the PUMA robot as a joystick, a forward displacement analysis is carried out for the completion of the work.

Serial robots are good examples for studying relative coordinate systems. Each link of the robot is usually

considered to have an attached local coordinate system. Encoders are associated with each of the joints. The joint angles can be found from the encoder readings. Applying the joint angles in a forward displacement analysis, location and orientation of the end-effector of the robot can be determined. The essence of the forward displacement analysis is to transform the coordinates of the end-effector associated with its local coordinate system into the base coordinate system. A systematic approach to implement this analysis is to line up the local transformation matrices between the base coordinate system and the end-effector coordinate system.

Although the forward displacement analysis is not difficult with any set of local coordinate systems, for practical reasons in this particular work, it is convenient to have the local coordinate system of each joint aligned with each axis of the factory-supplied PUMA 600 robot so that the measured joint angles can be utilized without further modifications.

Consider the situation depicted in Figure 2.3, where a point  $p$  is located in a space characterized by two different Cartesian coordinate systems. Suppose that the location of point  $p$  corresponding to coordinate system 1, denoted by  $P_1$ , and the displacement from coordinate 0 to coordinate system 1, denoted by  $D = (d_{x01} \ d_{y01} \ d_{z01})^T$ , are both known. The location of point  $p$  relative to coordinate system 0, given as  $P_0$ , can then be obtained by applying the concept of vector projection.

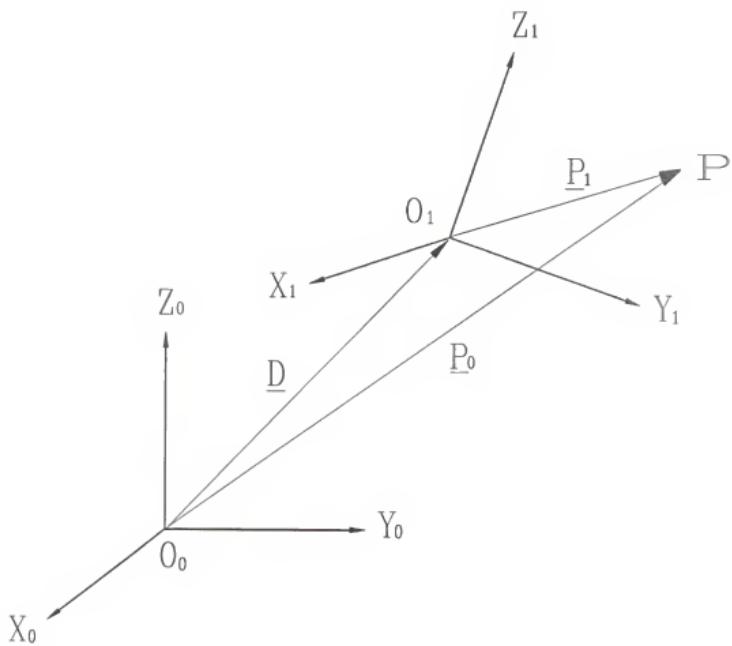


Figure 2.3 Relative Coordinate Systems of a Point

The transformation between  $P_1$  and  $P_0$  can be expressed by a homogeneous transformation matrix as

$$\begin{vmatrix} P_0 \\ 1 \end{vmatrix} = {}_0T^1 \begin{vmatrix} P_1 \\ 1 \end{vmatrix} \quad (2.1)$$

where

$${}_0T^1 = \begin{vmatrix} [R] & D \\ [0] & 1 \end{vmatrix}$$

and [0] is a  $1 \times 3$  null array,  $D$  is regarded as a translational vector between two coordinate systems, and  $[R]$  is a  $3 \times 3$  rotation matrix from coordinate system 1 to coordinate system 0, given as

$$[R] = \begin{vmatrix} i_1 \cdot i_0 & j_1 \cdot i_0 & k_1 \cdot i_0 \\ i_1 \cdot j_0 & j_1 \cdot j_0 & k_1 \cdot j_0 \\ i_1 \cdot k_0 & j_1 \cdot k_0 & k_1 \cdot k_0 \end{vmatrix}$$

Each column of  $[R]$  represents the perpendicular projection of a unit vector in coordinate system 1 onto coordinate system 0.  $[R]$  is an orthonormal matrix.

Following this convention, the local transformation matrix from coordinate frame  $i$  into coordinate frame  $i-1$  can be expressed as

$${}_{i-1}T^i = \begin{vmatrix} i_i \cdot i_{i+1} & j_i \cdot i_{i+1} & k_i \cdot i_{i+1} & d_{x,i-1i} \\ i_i \cdot j_{i+1} & j_i \cdot j_{i+1} & k_i \cdot j_{i+1} & d_{y,i-1i} \\ i_i \cdot k_{i+1} & j_i \cdot k_{i+1} & k_i \cdot k_{i+1} & d_{z,i-1i} \\ 0 & 0 & 0 & 1 \end{vmatrix}. \quad (2.2)$$

The forward transformation matrix between the end-effector coordinate system and the base coordinate system for a 6-DOF serial robot can be obtained by

$$T = {}_0T^1T^2T^3T^4T^5T^6. \quad (2.3)$$

Applying the kinematic parameters in Figure 2.1, the local transformation matrices are obtained as shown with illustrations in Table 2.3. The elements of transformation matrix T for the PUMA 600 robot are obtained as

$$T = \begin{vmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{vmatrix} = \begin{vmatrix} n & s & a & p \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (2.4)$$

where n, s, and a are unit vectors representing the orientation of the gripper, and p specifies the location of the origin of the gripper coordinate system in the base coordinate system, as shown in Figure 2.4, with

Table 2.3 Local Transformation Matrices of PUMA 600 Robot

$$\begin{aligned}
 {}_0T^1 &= \begin{vmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} & \text{Diagram: } & \begin{array}{c} Y_1 \\ | \\ X_0 \end{array} \xrightarrow{\theta_1} \begin{array}{c} Y_0 \\ | \\ X_0 \end{array} \\
 {}_1T^2 &= \begin{vmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} & \text{Diagram: } & \begin{array}{c} Z_1 \\ | \\ Y_2 \\ | \\ X_1 \end{array} \xrightarrow{\theta_2} \begin{array}{c} Z_1 \\ | \\ Y_2 \\ | \\ X_2 \end{array} \\
 {}_2T^3 &= \begin{vmatrix} c_3 & -s_3 & 0 & G \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & F \\ 0 & 0 & 0 & 1 \end{vmatrix} & \text{Diagram: } & \begin{array}{c} X_2 \\ | \\ Y_3 \\ | \\ Z_2 \\ | \\ X_3 \end{array} \xrightarrow{\theta_3} \begin{array}{c} X_2 \\ | \\ Y_3 \\ | \\ Z_3 \\ | \\ X_3 \end{array} \\
 {}_3T^4 &= \begin{vmatrix} c_4 & -s_4 & 0 & 0 \\ 0 & 0 & -1 & -H \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} & \text{Diagram: } & \begin{array}{c} Y_4 \\ | \\ Z_3 \\ | \\ X_3 \end{array} \xrightarrow{\theta_4} \begin{array}{c} Y_4 \\ | \\ Z_3 \\ | \\ X_4 \\ | \\ X_3 \end{array} \\
 {}_4T^5 &= \begin{vmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_5 & -c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} & \text{Diagram: } & \begin{array}{c} Z_4 \\ | \\ Y_5 \\ | \\ X_4 \end{array} \xrightarrow{\theta_5} \begin{array}{c} Z_4 \\ | \\ Y_5 \\ | \\ X_5 \\ | \\ X_4 \end{array} \\
 {}_5T^6 &= \begin{vmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & -1 & -L \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} & \text{Diagram: } & \begin{array}{c} Y_5 \\ | \\ Z_5 \\ | \\ X_5 \\ | \\ X_5 \end{array}
 \end{aligned}$$

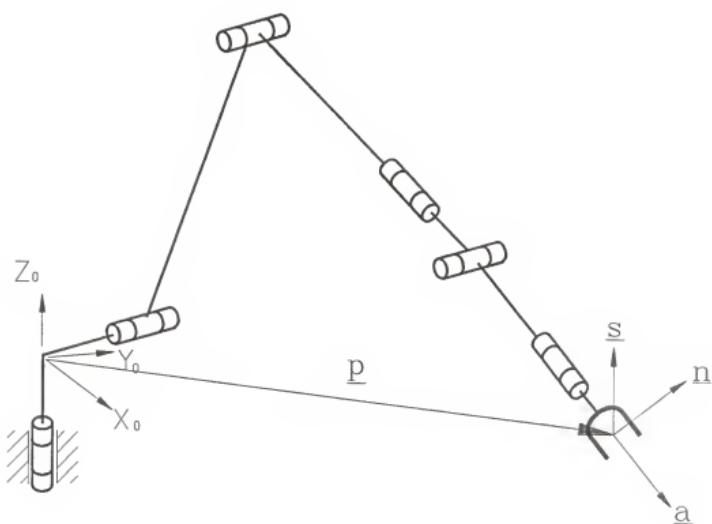


Figure 2.4 Tool Coordinates of the End-Effector

$$\begin{aligned}
n_x &= C_1 [C_{23}(C_4C_5C_6 - S_4S_6) - S_{23}S_5C_6] - S_1(S_4C_5C_6 + C_4S_6) \\
n_y &= S_1 [C_{23}(C_4C_5C_6 - S_4S_6) - S_{23}S_5C_6] + C_1(S_4C_5C_6 + C_4S_6) \\
n_z &= -[S_{23}(C_4C_5C_6 - S_4S_6) + C_{23}S_5C_6] \\
s_x &= C_1 [-C_{23}(C_4C_5S_6 + S_4C_6) + S_{23}S_5S_6] + S_1(S_4C_5S_6 - C_4C_6) \\
s_y &= S_1 [-C_{23}(C_4C_5S_6 + S_4C_6) + S_{23}S_5S_6] - C_1(S_4C_5S_6 - C_4C_6) \\
s_z &= S_{23}(C_4C_5S_6 + S_4C_6) + C_{23}S_5S_6 \\
a_x &= C_1(C_{23}C_4S_5 + S_{23}C_5) - S_1S_4S_5 \\
a_y &= S_1(C_{23}C_4S_5 + S_{23}C_5) + C_1S_4S_5 \\
a_z &= -S_{23}C_4S_5 + C_{23}C_5 \\
p_x &= L(C_1C_{23}C_4S_5 + C_1S_{23}C_5 - S_1S_4S_5) + HC_1S_{23} + GC_1C_2 - FS_1 \\
p_y &= L(S_1C_{23}C_4S_5 + S_1S_{23}C_5 + C_1S_4S_5) + HS_1S_{23} + GS_1S_2 + FC_1 \\
p_z &= L(-S_{23}C_4S_5 + C_{23}C_5) + HC_{23} - GS_2.
\end{aligned}$$

Note that only two among three orientation vectors,  $\underline{n}$ ,  $\underline{s}$ , and  $\underline{a}$ , are required. If necessary, the third can be obtained by a cross product of the other two vectors, for example,  $\underline{n} = \underline{s} \times \underline{a}$ .

## CHAPTER 3 SYSTEM DEVELOPMENT AND CONCEPTUAL DESIGN

In this project, the operator grasps the handgrip which is installed on the end-effector of the robot and desires that the robot respond to the instructions of his hand. A wrist force/torque sensor is used to sense the operator's application of wrenches. These sensed wrenches are further transformed into joint torques. A series of fundamental trial strategies is tested to control the PRJS.

Force information plays a significant role in this work. To clarify control strategies of robots, a brief review of force control is conducted in this chapter. The Jacobian transformation, which plays an important role in the PRJS, will be introduced in a practical sense. Also introduced is an analysis for the system stability which is helpful in the conceptual design stage.

### 3.1 Review of Force-Sensed Control of Robots

Force-controlled manipulators have been continually investigated in the literature [25, 26, 27, 28, 29, 30, 31, 32, 33]. Although numerous simulation results have been presented, there have been few published reports on the

experimental implementation of this type of controller. An exception to this statement is in the research of force-controlled direct drive arms. These have been recognized to be good only for testing control strategies as a result of their dominant inertia effects. In order to experimentally implement the force-controlled strategies for a robot, an appropriate robotic system and an appropriate on-line computation facility are needed.

Stemming from teleoperation, Inoue's selective joint torque control was an early development of a force-controlled manipulator which separated joints into position-controlled and torque-controlled ones [25]. This concept was generalized by Mason as the hybrid control theory based on the task geometry [26], and further experimentally verified by Raibert and Craig on two degrees of freedom for the 6-DOF Stanford robot [27]. The concept of hybrid control proposed software filters, which were formulated according to the constraints of task geometry, to distinguish between position-controlled and torque-controlled joints. In Raibert and Craig's experiment, the position commands were explicitly transformed into the joint torques via a specified stiffness matrix. They did report instability problems as observed in the experiment.

The control torques to drive each joint of a manipulator can be computed based on a suitable dynamic model, which is often referred to as computed-torque control. Lagrange formulation and Newton-Euler formulation are used to compute

the control torques [28, 29]. A feedback control scheme is essential in order to compensate for the unavoidable modeling errors. Owing to the complexity of the dynamic model of a robot, it is difficult to implement a real-time computed-torque control for a manipulator.

Khatib's COSMOS [30] system carries out the dynamically decoupled control torques in the so-called operational space, which is indeed a combination of position space and wrench space. Because of heavy computing operations, the software is executed in the NYMPH computer system, which includes five NSC 32016 processors and one MC 68010 processor connected with an Intel multibus.

Another approach is to determine the control torques by means of gain matrices with feedback position or velocity information. Whitney's damping control [31], Salisbury's stiffness control [32], and Hogan's impedance control [34] all fall into this category.

Differing from force control, most industrial robots are controlled with either position commands or velocity commands due to their easy implementation. In recent years, an interesting concept of the inner/outer control loop configuration, which can take advantage of the existing factory-supplied systems, has drawn more attention. With this configuration, the inner control loop can reject internal disturbances caused by the actuator since it is closed on the colocated joint sensor. The outer control loop can reject

the external disturbances originating from the interaction wrenches between the end-effector and the environment since it is closed on the force/torque sensor. Simons and Van Brussel in [35] and Maples and Becker in [36] all discussed the same concept.

The stability issue, which has been recognized recently [31, 37, 38, 39, 40, 41, 42, 43, 44, 45], is critical for force-controlled manipulators. Only limited efforts have been devoted to investigate the real causes of this problem. Whitney was the first to provide a stability analysis of a force-controlled manipulator and concluded that high bandwidth force control required a compliant sensor or environment [31]. Drawbacks to a soft sensor include retardation of the force response and deterioration of the accuracy of position control. Both An and Hollerback [37, 38] and Kazerooni et al. [42] showed that the tradeoff might be attributed to unmodeled dynamics. An and Hollerback also addressed the instabilities due to kinematic coordinate transformations (kinematic stability) and the stiffness effects with the environment (dynamic stability). Eppinger and Seering [39, 40, 41] carried out extensive stability analyses based on a series of mass-spring-dashpot single-axis models. They showed that stability problems arose due to non-colocation of the actuators and the sensors.

Kinesthetic instability was found to be another type of instability in hybrid control, first observed by Lipkin and

Duffy [43, 44, 45]. It results from strategies that are not invariant with respect to change of origin, basis, or scale. This instability phenomenon is associated with the problem of filtering spatial force and motion signals which can cause dimensional dissimilarity. Lipkin and Duffy determined the criteria to prevent kinesthetic instability was that the decomposition process be invariant with respect to a change of origin, basis, or scale.

Force control in robotic research has concentrated heavily on assembly and deburring tasks. Fine motions are all required and small control gains are sufficient for these tasks. For the reason of simplicity, the analyzed systems are often modeled as single DOF systems.

Other than the assembly and deburring tasks, Khatib and Burdick in [30] introduced "zero force" control on the Z-direction of a PUMA 560 robot. The robot dynamics were mathematically decoupled to demonstrate the computed-torque control. Maples and Becker in [36] introduced their "teach mode" on a planar AdeptOne robot, in which the robot was expected to respond to external forces. The external forces were transformed into accelerations by selected compensators with the gain of 0.005. The above two control algorithm experiments were developed in Cartesian coordinates. Since these operations were by-products of an assembly operation, they showed the ability of the robot to comply to external

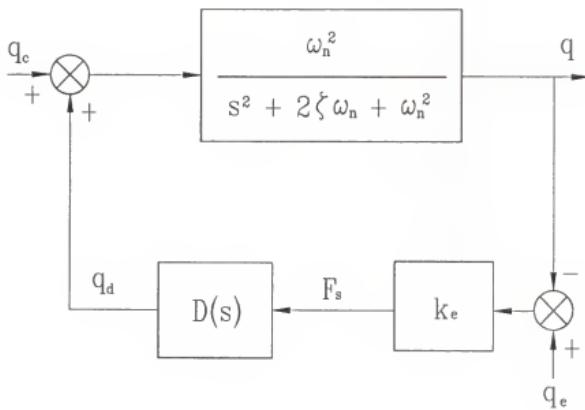
force commands but did not report further consequences or developments.

### 3.2 Joystick Subsystem

The factory-supplied PUMA robot, when the servo is on, will remain at the commanded positions and orientations even if external wrenches are applied. The present work adds an external loop around the original control loop to accommodate the operator's application of wrenches. In this way, the servo of the original system is being used. The system is shown in Figure 3.1.

A commercially available wrist force/torque sensor, the FT-15/50 supplied by Lord, is used in the PRJS. The sensor comes with a controller unit, a sensor transducer unit, a preprocessor unit, and cables to connect them together. It can measure forces up to 15 lb<sub>f</sub> and torques up to 50 lb<sub>f</sub>-in with resolutions of 0.2 oz in the X- and the Y-directions of the sensor coordinate, 0.6 oz in the Z-direction of the sensor coordinate, and 0.4 oz-in for torques. The sampling rate of this system can be as high as 104 Hz [46].

Once the force/torque sensor is installed, a design of a control scheme is needed for the PRJS. The control scheme should be general enough so that it can be applied with only minor modifications to a variety of robots. It is even better if the control scheme can be carried out without detailed



$q$  : Output joint angle

$q_e$  : Externally applied movement

$q_c$  : Commanding joint angle

$q_d$  : Feedback delta joint angle

$\omega_n$  : Natural frequency

$k_e$  : Environmental stiffness constant

$F_s$  : Sensed force/torque

$\zeta$  : Damping ratio

Figure 3.1 System Configuration of a Single Degree-of-Freedom Joystick Subsystem

knowledge of the joint controller, because this information is usually proprietary.

Note that the robot is inherently a low bandwidth flexible system. When the force/torque sensor is attached to the end of this flexible structure, a high feedback gain will produce instability. Therefore, only limited gains can produce a stable system with the application of a wrist force/torque sensor. Furthermore, the measured wrenches, which are in Cartesian coordinates, need to be transformed into joint torques.

In order for the TANDY 4000 computer to take over control of the robot system, an economical approach is to let the two ribbon cables with 40-pin connectors, which connect the AIB to the DRV-11 parallel interface, be redirected to the PC-11 parallel interface in the TANDY 4000 computer. The TANDY 4000 computer then emulates the control protocols just as the LSI-11/02 CPU does. Under this circumstance, the so-called DEC-components in the PUMA controller are all isolated.

Once the connection is completed, the TANDY 4000 computer is ready to communicate with the joint microprocessors of the PUMA robot. The procedures INITIAL and CALIB in the TANDY 4000 computer need to be executed so that the computer and the joint microprocessors can recognize the same measurement system.

Figure 3.2 shows the block diagram of the PRJS. The TANDY 4000 computer in the system has an 80386 microprocessor,

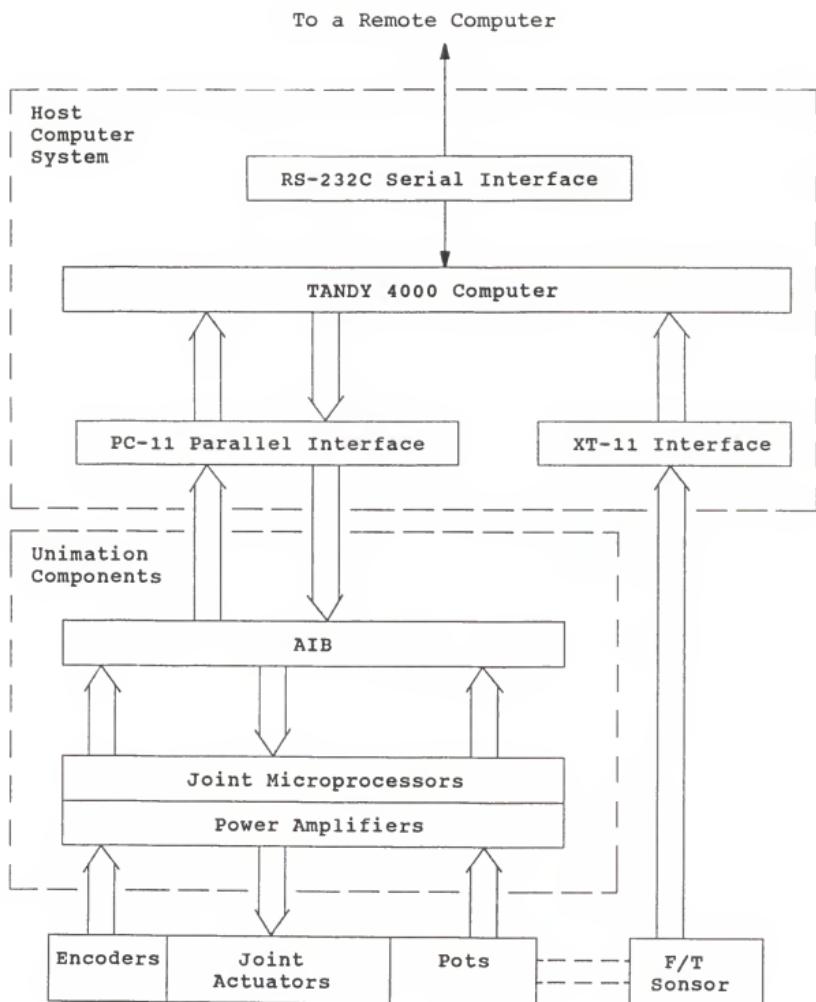


Figure 3.2 Block diagram of the PUMA joystick subsystem

an 80387 coprocessor, a 16 MHz system clock, and an RS-232C serial port. Two parallel interfaces, one XT-11 card and one PC-11 card, which emulate the DEC DRV-11 parallel interface, are also installed in the system. The XT-11 card reads the wrench information from the FT-15/50 force/torque sensor once every cycle. The PC-11 card takes care of the communication between the TANDY computer and the AIB in the PUMA controller. With this configuration, the DEC-11/02 CPU of the factory-supplied PUMA system is abandoned and thus the VAL codes in the EPROM are not accessible. The TANDY computer must take charge of all the command generations and the I/O with the joint microprocessors of the robot.

The next paragraph describes some signal flows at the chip level of the PC-11 interface. They are helpful only for those who work on the hardware signal checking.

In this subsystem, the addresses of the PC-11 interface are selected as

Control Status Register =  $300_{16}$ ,

Output register =  $302_{16}$ , and

Input Register =  $304_{16}$ .

CSR0, bit 4 of the control status register of the PC-11 interface [47], is used to direct the data flow. When CSR0 is zero, data written to the output register are taken as a command. When CSR0 is one, data are taken as the data

corresponding to the previously issued command and can be written to the output register or be read from the input register, depending on the issued command. During an output cycle, a NEW DATA READY pulse from the PC-11 interface is generated to inform the joint microprocessors in the PUMA controller of data transfer. The trailing edge of this positive-going pulse passes data into the joint microprocessors. During an input cycle, when a read of the lower byte is performed, a positive-going DATA TRANS pulse is generated which informs the joint microprocessors of the PUMA controller that data have been accepted. The trailing edge of the pulse indicates that data transfer has been completed.

The status of REQ A, which is readable from bit 3 of the control status register of the PC-11 interface, indicates whether the signal is ready to be transferred or not. Data should not be read from the input register or written to the output register before REQ A bit is set to be one.

Special care must be taken in using the PC-11 interface in order to have proper communications. The design of the PC-11 interface uses an 8-bit architecture to emulate the DRV-11 interface which has a 16-bit architecture. After a command character is written to port  $302_{16}$ , a dummy character should be written to port  $303_{16}$  in order to keep synchronization of communication.

To execute a program involving low-level I/O functions on a personal computer, a difficult situation comes about when

the executed program is not completed properly. The computer system will become trapped and it must be rebooted. This is certainly not convenient for developing and testing programs. The language available in the TANDY computer is Turbo-C 2.0 from Borland company running under MS-DOS. The Turbo-C environment provides a good solution with the Ctrl-Break key which allows the computer to get back to the program without rebooting.

The Turbo-C environment, however, does have some drawbacks. When a program is executed, all the parameters are allocated to certain memory locations. If the program is not modified, Turbo-C will not recompile it. Under this circumstance, the content of the allocated memory will not be updated when the same program is executed again, regardless of the initialization statements. An extra key stroke, Ctrl-F2, to reset the program is required. It releases the allocated memory from the previous run. This "reset program" key is especially necessary when the computer system is communicating with another computer in order to eliminate the unwanted characters in the buffer.

Effective control of the PUMA robot as a joystick requires execution of all actions within the 28 msec cycle time. In this time period, the TANDY computer reads the encoder codes from the PUMA robot, reads the wrench information from the wrist force/torque sensor, and generates the position commands according to the collected information

and the control scheme. Consequently, the program codes must be simple enough for real-time execution on the TANDY computer. Those statements for I/O functions are required and cannot be further optimized within the current working environment, unless an assembly language compiler is available. Other computation codes should be watched carefully. Although data structure represents a clear concept of programming and the pointer is an efficient tool in the C language, all parameters in the control program are kept constant as long as possible. Also, "assign" statements which can be eliminated are removed so that the condensed program codes are obtained. In this way, the program codes are optimized for execution speed.

For now, an RS-232C port in the TANDY computer serves as a communication port when the connection between the PRJS and a remote device is considered. With the assistance of this serial port, joint commands can be sent to the remote device and the reflecting force information can be fed back to the PRJS from a remote device.

### 3.3 Application of the Jacobian Matrix

Considering a system in static equilibrium, the principle of virtual work can be expressed by

$$\underline{F} \cdot \delta \underline{x} = \underline{\Gamma} \cdot \delta \underline{\theta} \quad (3.1)$$

where  $\underline{F}$  is a  $6 \times 1$  vector of wrenches acting on the handgrip,  $\delta \underline{x}$  is a  $6 \times 1$  vector of generalized infinitesimal displacements of the handgrip,  $\underline{\Gamma}$  is a  $6 \times 1$  vector of joint torques,  $\delta \underline{\theta}$  is a  $6 \times 1$  vector of infinitesimal joint displacements, and " $\cdot \cdot$ " denotes the inner product of two vectors. Expression (3.1) can also be written as

$$\underline{F}^T \delta \underline{x} = \underline{\Gamma}^T \delta \underline{\theta} \quad (3.2)$$

where the superscript " $T$ " denotes the matrix transpose. The definition of the Jacobian is

$$\delta \underline{x} = [\text{Jacobian}] \delta \underline{\theta}. \quad (3.3)$$

Substituting equation (3.3) into equation (3.2) yields

$$\underline{F}^T \delta \underline{x} = \underline{F}^T [\text{Jacobian}] \delta \underline{\theta} = \underline{\Gamma}^T \delta \underline{\theta}. \quad (3.4)$$

The relation between wrenches and joint torques can be obtained by transposing equation (3.4) as

$$\underline{\Gamma} = [\text{Jacobian}]^T \underline{F}. \quad (3.5)$$

Be aware of the fact that the Jacobian is a function of joint angles. Hence, different Jacobian matrices will result with respect to different reference coordinates. In addition,

the Jacobian matrix needs to be updated whenever the configuration of the robot is changed. This can introduce a heavy computational load.

Several researchers have recognized that the simplest computations of the Jacobian matrix occur for the coordinate system whose origin is located at the intersection of three wrist joints. This is true for robots such as the PUMA [48, 49, 50, 51, 52]. Amongst those researchers, Hunt introduces his selection of a referencing coordinate system ( $O_4X_4Y_4Z_4$ ) as shown in Figure 3.3 [51]. In this coordinate system,  $O_4Y_4$  is made to be parallel to  $O_1Y_1$ . Based on screw theory, the Jacobian matrix with minimum mathematical operations can be obtained by inspecting the figure.

Conceptually, screw theory is considered as an elegant tool to investigate kinestatics (kinematics and statics). A practical implementation of screw theory is desirable for the application of the Jacobian matrix. For those readers who are interested in the theoretic background of screw theory, a list of references [such as 43, 44, 53, 54, 55] is helpful.

A line in ( $O_jX_jY_jZ_j$ ) space can be represented as

$$\underline{S}_j = \begin{vmatrix} S_j \\ \underline{x}_j \times \underline{S}_j \end{vmatrix} = (L_j, M_j, N_j; P_j, Q_j, R_j)^T, \quad (3.6)$$

where the symbol "x" denotes the cross product of vectors,  $S_j$  denotes a 3x1 vector representing the direction of  $\underline{S}_j$ ,  $\underline{x}_j$

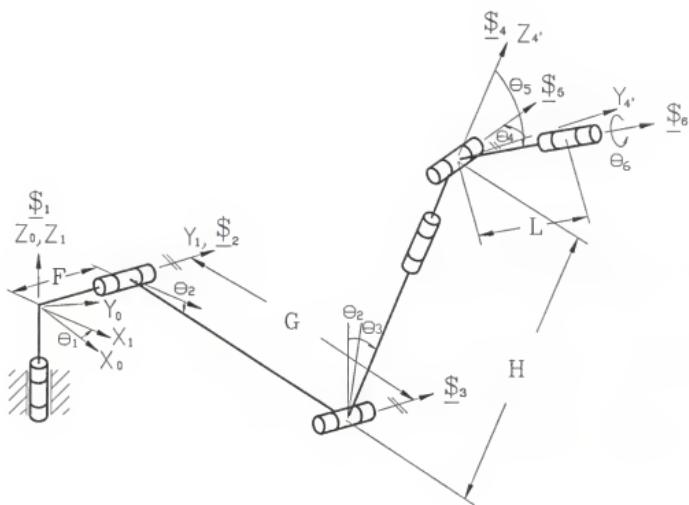


Figure 3.3 PUMA 600 Robot with Coordinate System Suggested by Dr. Hunt [51]

denotes a  $3 \times 1$  vector pointing from the origin  $O_j$  to a point on the line, and  $\underline{r}_j \times \underline{s}_j$  represents the moment of the line with respect to the origin  $O_j$ .

When a screw is designated for each revolute joint, the Jacobian matrix of a 6-DOF serial robot can be expressed as

$$J = \begin{vmatrix} \underline{s}_1 & \underline{s}_2 & \underline{s}_3 & \underline{s}_4 & \underline{s}_5 & \underline{s}_6 \\ \underline{r}_1 \times \underline{s}_1 & \underline{r}_2 \times \underline{s}_2 & \underline{r}_3 \times \underline{s}_3 & \underline{r}_4 \times \underline{s}_4 & \underline{r}_5 \times \underline{s}_5 & \underline{r}_6 \times \underline{s}_6 \end{vmatrix}. \quad (3.7)$$

When the coordinate frame is chosen at the interaction of the last three joints, the cross product terms  $\underline{r}_4 \times \underline{s}_4$ ,  $\underline{r}_5 \times \underline{s}_5$ , and  $\underline{r}_6 \times \underline{s}_6$  all become zero vectors and the number of elements to be calculated in the Jacobian matrix is reduced.

The Jacobian matrix associated with the coordinate frame  $(O_4X_4Y_4Z_4)$  is obtained as

$$J = \begin{vmatrix} -S_{23} & 0 & 0 & 0 & -S_4 & S_5 C_4 \\ 0 & 1 & 1 & 0 & C_4 & S_5 S_4 \\ C_{23} & 0 & 0 & 1 & 0 & C_5 \\ -F C_{23} & (H + G S_3) & H & 0 & 0 & 0 \\ (H S_{23} + G C_2) & 0 & 0 & 0 & 0 & 0 \\ -F S_{23} & -G C_3 & 0 & 0 & 0 & 0 \end{vmatrix}. \quad (3.8)$$

In order to apply this Jacobian matrix with wrenches acting at the end-effector, the transpose of this Jacobian matrix must be post-multiplied by a  $6 \times 6$  correction matrix

$$\Delta = \begin{vmatrix} [0_3] & [I_3] \\ [I_3] & [0_3] \end{vmatrix}, \quad (3.9)$$

where  $[0_3]$  is the  $3 \times 3$  null matrix and  $[I_3]$  is the  $3 \times 3$  identity matrix. Thus,

$$J^T \Delta = \begin{vmatrix} -FC_{23} & (HS_{23}+GC_2) & -FS_{23} & -S_{23} & 0 & C_{23} \\ (H+GS_3) & 0 & -Gc_3 & 0 & 1 & 0 \\ H & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -S_4 & C_4 & 0 \\ 0 & 0 & 0 & S_5 C_4 & S_5 S_4 & C_5 \end{vmatrix}.$$

Furthermore, wrenches are measured at locations different from the place where the Jacobian matrix is computed. We also need the transformation between these two coordinate systems.

Given two coordinate systems,  $(O_i X_i Y_i Z_i)$  and  $(O_j X_j Y_j Z_j)$ , the coordinates of  $O_j$  with respect to  $(O_i X_i Y_i Z_i)$  can be expressed by a vector  $(x_{ij}, y_{ij}, z_{ij})^T$ . A screw in the  $(O_j X_j Y_j Z_j)$  system can then be transformed into the  $(O_i X_i Y_i Z_i)$  system by the transformation matrix

$$[\$] = \begin{vmatrix} [R] & [0_3] \\ [D][R] & [R] \end{vmatrix} \quad (3.10)$$

where  $[R]$  is a  $3 \times 3$  orthogonal rotation matrix, given, for example, as

$$[R] = \begin{vmatrix} C_i & S_i & 0 \\ S_i & C_i & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

in the case when  $O_j Z_j$  and  $O_i Z_i$  are parallel to each other and the orientation of  $(O_j X_j Y_j Z_j)$  with respect to  $(O_i X_i Y_i Z_i)$  is specified by an angle  $\theta_i$  using the right-hand rule. The matrix  $[D]$  is a  $3 \times 3$  skew-symmetric matrix in terms of the coordinates of  $O_j$  relative to the  $(O_i X_i Y_i Z_i)$  system, given as

$$[D] = \begin{vmatrix} 0 & -z_{ij} & y_{ij} \\ z_{ij} & 0 & -x_{ij} \\ -y_{ij} & x_{ij} & 0 \end{vmatrix}.$$

Thus, a pure rotational transformation matrix corresponding to axis A with an angle  $\theta$  can be defined by a  $6 \times 6$  matrix as

$$[\text{Rot}(A, \theta)] = \begin{vmatrix} [R] & [0_3] \\ [0_3] & [R] \end{vmatrix} \quad (3.11)$$

and a pure translational transformation matrix corresponding to axis A with a distance L,

$$[\text{Tran}(A, L)] = \begin{vmatrix} [0_3] & [0_3] \\ [D] & [0_3] \end{vmatrix}. \quad (3.12)$$

The measured wrenches are mapped from coordinates  $(O_6X_6Y_6Z_6)$  to  $(O_4X_4Y_4Z_4)$  by applying these transformation matrices. That is

$$\underline{F} = T\underline{F}_4 \quad (3.13)$$

with

$$T = [\text{Rot}(Z_4, \theta_4)] [\text{Rot}(Z_5, \theta_5)] [\text{Rot}(Z_6, \theta_6)] [\text{Tran}(Z_6, l)]$$

or

$$T = \begin{vmatrix} [A] & [0_3] \\ [B] & [A] \end{vmatrix}$$

where

$$[A] = \begin{vmatrix} (C_6C_5C_4 - S_6S_4) & -(S_6C_5C_4 + C_6S_4) & S_5C_4 \\ (C_6C_5S_4 + S_6C_4) & (-S_6C_5S_4 + C_6C_4) & S_5S_4 \\ -C_6S_5 & S_6S_5 & C_5 \end{vmatrix}$$

$$[B] = \begin{vmatrix} -L(S_6C_5C_4 + C_6S_4) & -L(C_6C_5C_4 - S_6S_4) & 0 \\ L(-S_6C_5S_4 + S_6C_4) & -L(C_6C_5S_4 + S_6C_4) & 0 \\ LS_6S_5 & LC_6S_5 & 0 \end{vmatrix}$$

and  $[0_3]$  is the  $3 \times 3$  null matrix. Hence, the joint torques can be obtained by

$$\underline{\Gamma} = [J^T \Delta] T \underline{F}_4. \quad (3.14)$$

The order of computations for the elements of the transformation matrix is also important. With an appropriate arrangement, computational operations can be reduced. For example, after the elements in the first column of matrix [A] are calculated, the elements in the second column of matrix [B] can be obtained by multiplying L with them.

### 3.4 Stability Analysis

The approaches to evaluate the stability of system dynamics are

- 1) Routh criterion, if no delay is involved,
- 2) Bode plot, also if no delay is involved, and
- 3) Nyquist plot, if time delay is involved.

The Routh criterion may give a closed form for the stability boundary, depending on the order and the complexity of the characteristic equation obtained from the closed-loop transfer function of the system. A Bode plot shows the stability of the system based on the open-loop transfer function. Since 0 dB<sup>1</sup> corresponds to a magnitude of 1, the gain margin is the number of dB of the magnitude  $|GH(j\omega)|$  below 0 dB at the phase crossover frequency  $\omega_c$ , where  $\arg(GH(j\omega_c)) = 180^\circ$ . The phase margin is the number of

---

<sup>1</sup>  $dB = 20\log|GH(j\omega)|$ .

degrees of  $\arg(GH(j\omega))$  above  $-180^\circ$  at the gain crossover frequency  $\omega_1$ , where  $|GH(j\omega_1)| = 1$ . A Bode plot gives the gain margin and phase margin of the system. Positive gain margin and positive phase margin assure stability of the closed-loop system. When a time delay is involved, a closed form for the stability boundary is hardly ever obtained. Alternatively, the stability can be determined from a Nyquist plot. Actually, stability of the system can be decided by a glance at the Nyquist plot.

Industrial robots are often designed with second-order critical damping systems. A single degree-of-freedom robotic system with a force feedback loop looks like Figure 3.1. The diagram can further be arranged to be a commonly-used negative feedback system, as far as the open-loop transfer function,  $q_d/q_c$ , and the closed-loop transfer function,  $q/q_c$ , are concerned. An equivalent system is shown in Figure 3.4 and will be used for stability analysis of the system.

Consider the compensator as

$$D(s) = \frac{1}{Bs + K} \quad (3.15)$$

where  $B$  is the damping constant and  $K$  is the stiffness constant; then,

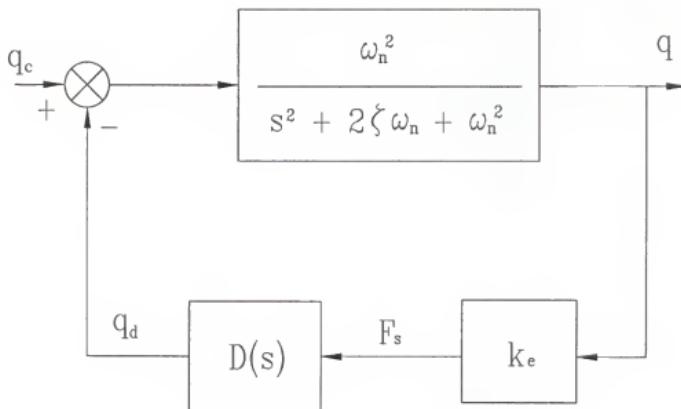


Figure 3.4 Equivalent Configuration of a Single Degree-of-Freedom Joystick Subsystem

$$G_{op}(s) = \frac{q_d}{q_c} = \frac{\omega_n^2 K_e}{(Bs + K)(s^2 + 2\zeta\omega_n s + \omega_n^2)},$$

$$G_{cl}(s) = \frac{q}{q_c} = \frac{\omega_n^2 K_e}{(Bs + K)(s^2 + 2\zeta\omega_n s + \omega_n^2) + \omega_n^2 K_e}.$$

The characteristic equation looks like

$$Bs^3 + (K + 2\zeta\omega_n B)s^2 + (\omega_n^2 B + 2\zeta\omega_n K)s + (K + K_e)\omega_n^2.$$

The stability conditions, according to the Routh criterion, require

$$B > 0,$$

$$K + 2\zeta\omega_n B > 0,$$

$$(K + 2\zeta\omega_n B)(\omega_n^2 B + 2\zeta\omega_n K) > (K + K_e)\omega_n^2 B,$$

or

$$K > -\zeta\omega_n B + (\zeta^2\omega_n^2 B^2 - \omega_n^2 B^2 + 0.5\omega_n K_e B)^{\frac{1}{2}}.$$

For an industrial robot which has a damping ratio of one,

$$K > -\omega_n B + (0.5\omega_n K_e B)^{\frac{1}{2}}. \quad (3.16)$$

The relation between the damping constant B and the stiffness constant K can be plotted as Figure 3.5 once  $K_e$  is picked. As can be seen from the figure, for a stiffer environment where

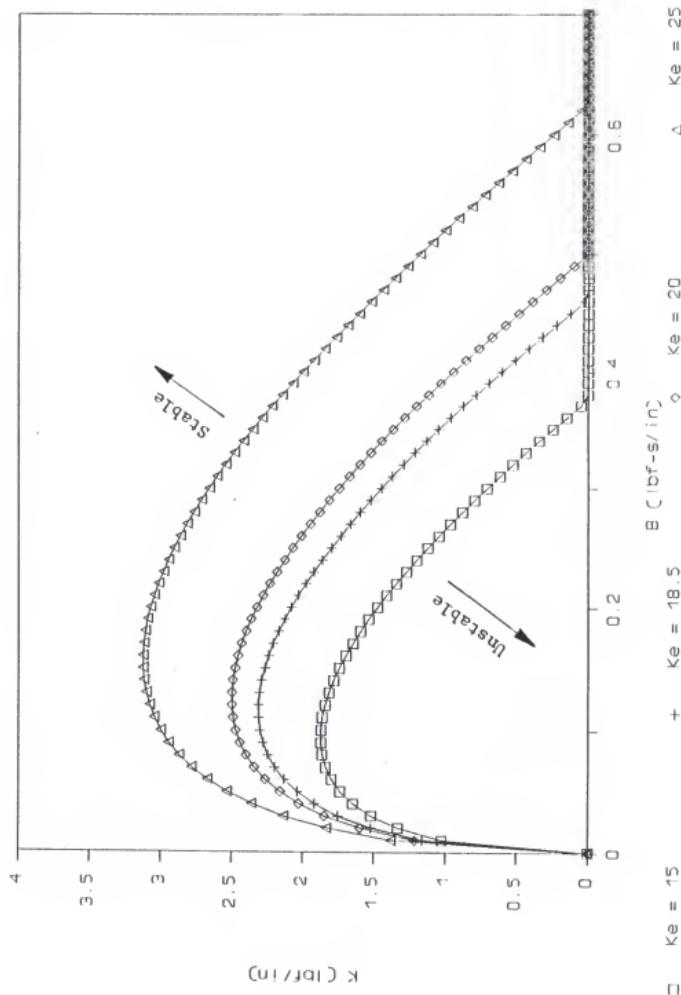


Figure 3.5 Damping-Stiffness Curves with Variable  $K_s$ ,  $\xi = 1$

$K_e$  is large, the stable zone becomes smaller. After the stability boundary curves pass through their peak values of  $K$ ,  $K$  becomes more or less proportional to  $-B$ . The performing point must be chosen within the stable zone. It is understandable from the figure that for a performing point with a small value of  $B$ , a very large value of  $K$  is required. This is certainly not good for the application of a joystick system. Since a good amount of  $B$  is always needed to damp out the undesired energy for a joystick system, a fixed small value of  $K$  and a large value of  $B$  are selected. Figure 3.6 shows a Bode plot with different values of  $B$  in the case of  $K_e = 18.5 \text{ lb}_f/\text{in}$ ,  $K = 1 \text{ lb}_f/\text{in}$ . The results confirm what is shown in Figure 3.5 where the stable zone occurs for  $B$  larger than  $0.4 \text{ lb}_f\text{-s/in}$ .

It was also observed that, as long as the square root term in equation (3.16) is kept positive, the damping ratio has little effect on the system stability. A case with  $K_e = 20 \text{ lb}_f/\text{in}$  is plotted as Figure 3.7 for variable damping ratios. As can be seen, the stability boundary curves change slightly, especially when a small value of  $K$  is selected. Another interesting observation is that a string with  $K = 0$  always passes through the same points of the set of curves which have the same value of  $K_e$ .

Although the PRJS has only one CPU, the program codes have been condensed and can be executed completely in every cycle. The time delay is therefore not significant. The

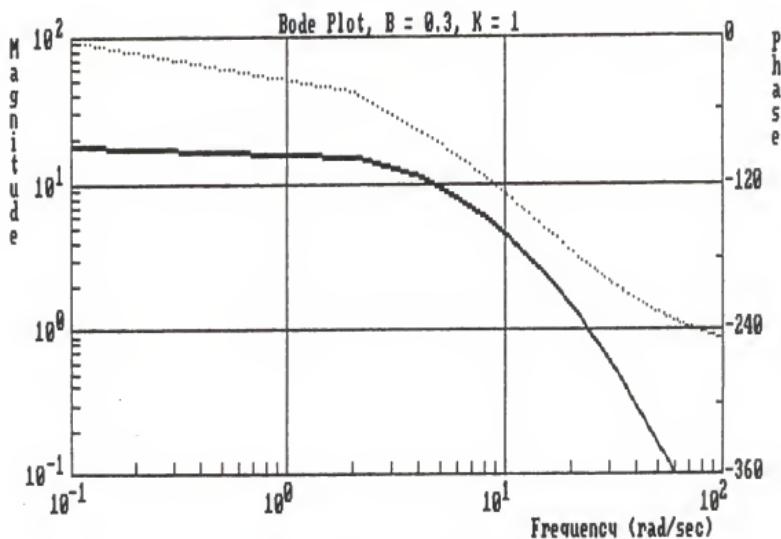


Figure 3.6(a) Bode Plot,  $B = 0.3, K = 1$

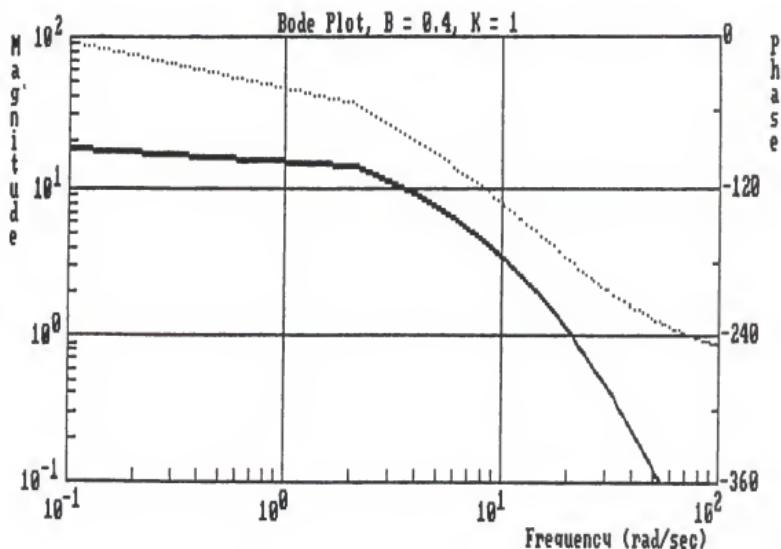
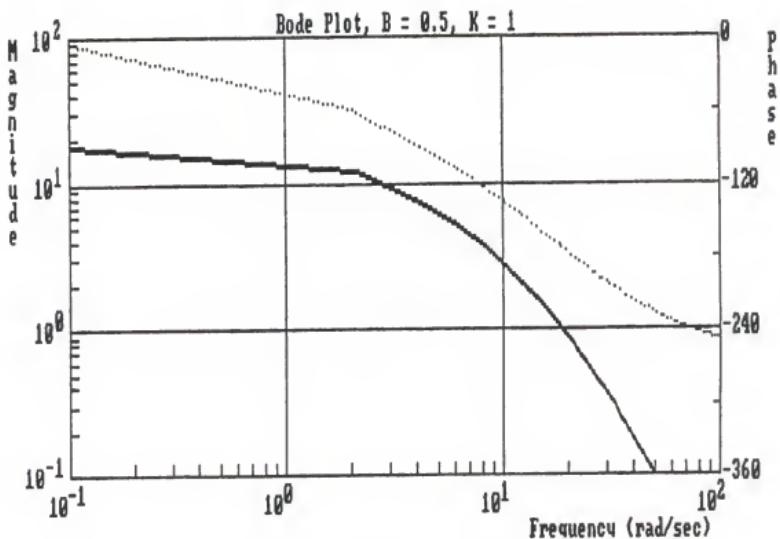
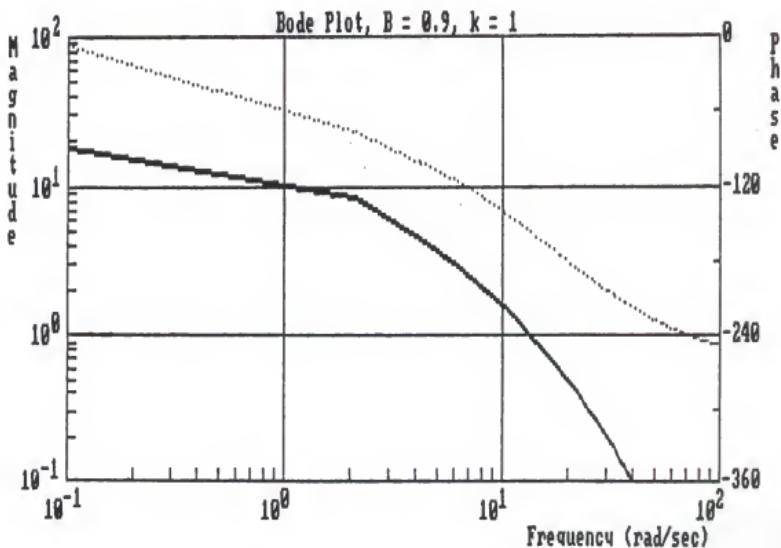


Figure 3.6(b) Bode Plot,  $B = 0.4, K = 1$

Figure 3.6(c) Bode Plot,  $B = 0.5, K = 1$ Figure 3.6(d) Bode Plot,  $B = 0.9, K = 1$

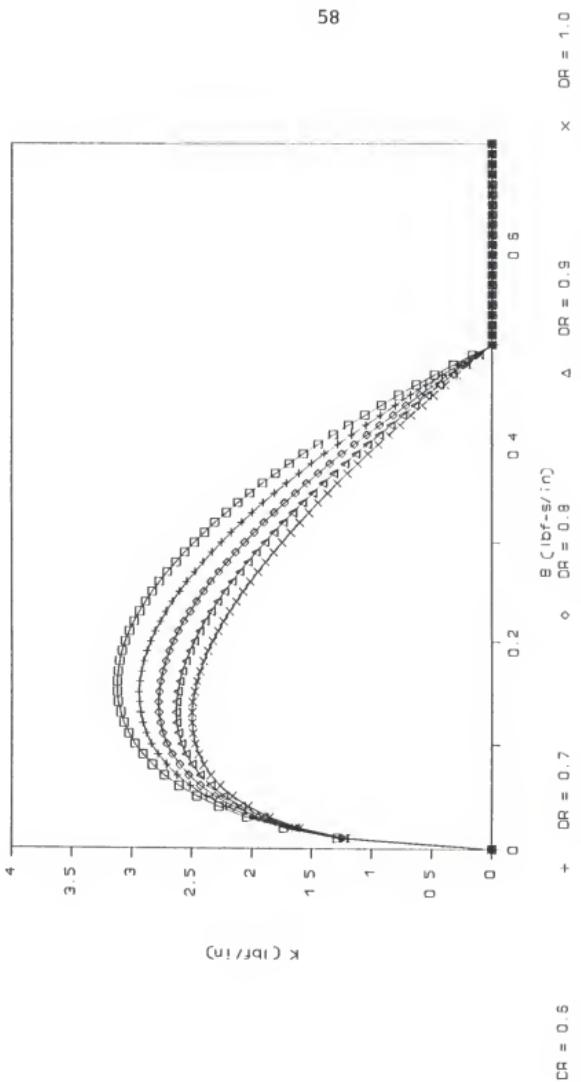


Figure 3.7 Damping-Stiffness Curves with Variable Damping Ratio,  $K_e = 20 \text{ lb/in}$

influence of time delay to the control system is, however, explored as a reference.

The open-loop transfer function of the model with a time delay  $T_d$  is

$$G_{op}(s) = \frac{q_d}{q_c} = \frac{\omega_n^2 K_e e^{-sT_d}}{(Bs + K)(s^2 + 2\zeta\omega_n s + \omega_n^2)},$$

By setting the phase margin to zero, a relation between impedance control design parameters  $K$  and  $B$  is provided. This relation is given by two nonlinear equations in the frequency  $\omega$  as

$$\omega_n^2 K_e = [((\omega_n^2 - \omega^2)^2 + (2\zeta\omega_n\omega)^2)(K^2 + B^2\omega^2)]^{1/4},$$

$$-\pi = -\tan^{-1}\left(\frac{2\zeta\omega_n\omega}{\omega_n^2 - \omega^2}\right) - \tan^{-1}\left(\frac{B\omega}{K}\right) - \omega T_d.$$

Owing to the nonlinearity, an analytic solution needs to be obtained by numerical methods.

Nyquist plots for the systems with and without delay are plotted as Figure 3.8 and Figure 3.9, respectively. It is found that the time delay has quite an influence on system stability. Using the same case of the previous example,  $K_e = 18.5 \text{ lb}_v/\text{in}$ ,  $K = 1 \text{ lb}_v/\text{in}$ ,  $B$  is now shifted from  $0.4 \text{ lb}_v\text{-s/in}$  to  $0.9 \text{ lb}_v/\text{in}$  in order to keep the system stable. A larger value

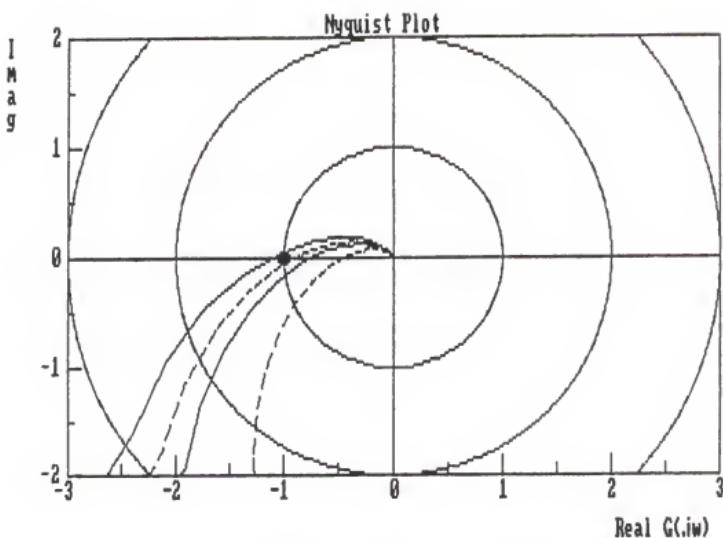


Figure 3.8 System Nyquist Plot with no Delay

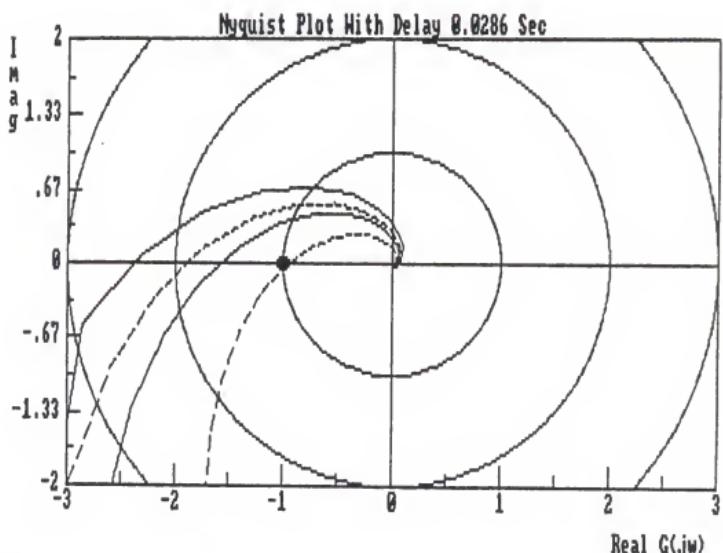


Figure 3.9 System Nyquist Plot with a Delay of 0.028 Sec

of B is required for the system with a longer delay. Bode plots for the systems with a time delay of one cycle are shown in Figure 3.10 for comparison.

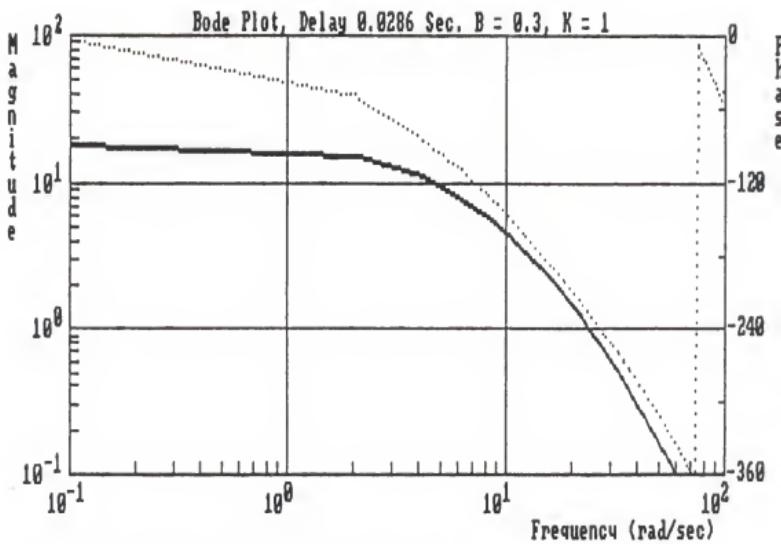


Figure 3.10(a) Bode Plot with a Delay of 0.028 Sec

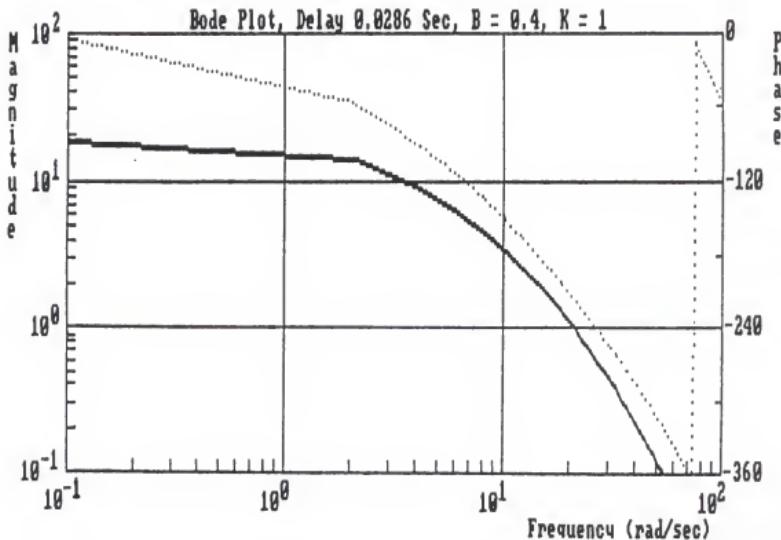


Figure 3.10(b) Bode Plot with a Delay of 0.028 Sec

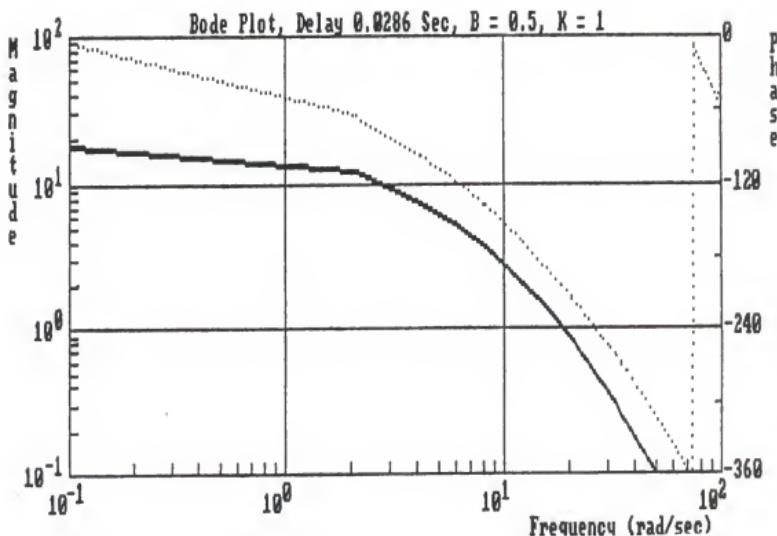


Figure 3.10(c) Bode Plot with a Delay of 0.028 Sec

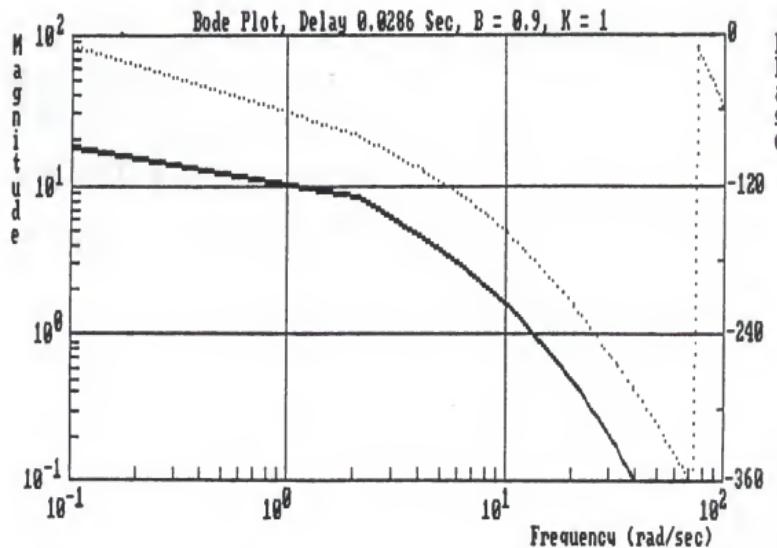


Figure 3.10(d) Bode Plot with a Delay of 0.028 Sec

## CHAPTER 4 EXPERIMENTAL IMPLEMENTATION AND RESULTS

Primitively, the PUMA robot can be controlled by either motor-current commands or joint-angle commands. Since most robot-control literature refers to torque control, and actuator torque is a function of motor current, motor-current commands were experimented to control the robot. Experimental results showed that this approach was less than satisfactory because of the dominant effects of Coulomb and viscous friction. Therefore, joint-angle commands were adopted in this project.

There are many advantages to controlling a robot system with joint-angle commands. None of the kinematic singularities of the robot arise, and fewer computations are involved, since the transformations between the world coordinate system and the joint coordinate system are eliminated. Physical joint angles are attractive for use when controlling a robot because of their intuitive nature. However, it is far more efficient to use encoder codes because they are the default measurement system of the joint microprocessors. If joint angles are utilized, they must be converted into encoder codes before being sent to the joint

microprocessors. Feedback encoder codes must also be converted into joint angles to close the control loop. This certainly costs more computations than in the case of using joint-encoder codes. Furthermore, joint angles are normally carried as "real" numbers, whereas encoder codes are used in "integer" form. Conversion procedures can produce truncation errors. In this chapter, joint angles, which indeed relate to encoder codes, are used for illustration.

#### 4.1 On-Desk Sensor

At the first stage of this project, the FT-15/50 force/torque sensor was placed on the desk as a commanding device for the PUMA 600 robot. This experiment gave a taste for driving the robot with the application of wrenches on a remote device. It verified that no dynamic modes were involved when integrating a signal into a system with no physical attachment to the system. This statement is true even if a force/torque sensor is used. Telepresence was arranged in this experiment through geometrical transformations. Also shown was that a different approach for stiffness control was feasible.

Experience shows that it is desirable to arrange the coordinate system of the force/torque sensor on the desk to align with the world coordinate system of the PUMA robot so that the wrenches acting on the sensor can move the end-effector of the PUMA robot in the same direction. This sense

of telepresence is simply obtained by the geometric transformation between the sensor coordinate system and the world coordinate system. For example, in order to respond the wrenches applied to the coordinate frame attached to the center of joint four, the rotation matrix can be expressed by, as seen in Figure 4.1,

$$T = [\text{Rot}(Z, \theta_1)][\text{Rot}(Y, \theta_{23})] \quad (4.1)$$

where

$$[\text{Rot}(Z, \theta_1)] = \begin{vmatrix} C_1 & S_1 & 0 & 0 & 0 & 0 \\ -S_1 & C_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & C_1 & S_1 & 0 \\ 0 & 0 & 0 & -S_1 & C_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix}$$

and

$$[\text{Rot}(Y, \theta_{23})] = \begin{vmatrix} C_{23} & 0 & -S_{23} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ S_{23} & 0 & C_{23} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{23} & 0 & -S_{23} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & S_{23} & 0 & C_{23} \end{vmatrix}.$$

The measured wrenches need to be transformed into the joint torques in the world coordinates by

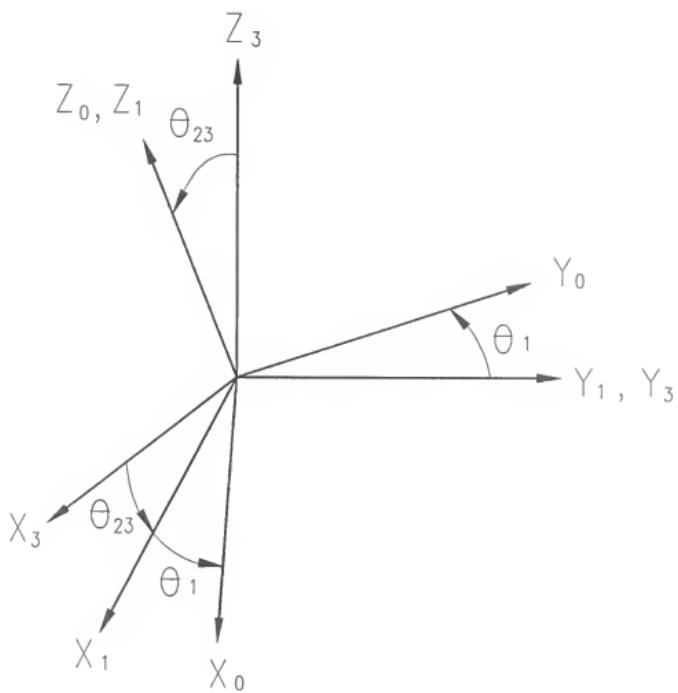


Figure 4.1 Telepresence Coordinate Transformation of the On-Desk Sensor

$$\boldsymbol{\Gamma} = [\mathbf{J}^T \boldsymbol{\Delta}] \mathbf{T} \mathbf{F}_e. \quad (4.2)$$

Applying these resulting torques with stiffness control, which is a special case of impedance control, the angular displacements to drive each joint of the robot are obtained.

The term "stiffness control" in this experiment is different from its classical meaning. Salisbury's definition of stiffness control [29] is a typical one in the literature of robotic control, which defines a stiffness matrix based on Cartesian coordinates. That is, the stiffness matrix  $\mathbf{K}$  in Cartesian coordinates is a diagonal matrix. The formulation of the joint torque can then be obtained as

$$\boldsymbol{\Gamma} = \mathbf{J}^T \cdot \mathbf{K} \cdot \mathbf{J} \cdot d\mathbf{q} = \mathbf{K}_\theta \cdot d\mathbf{q}, \quad (4.3)$$

where  $\mathbf{K}_\theta$  denotes the torsional stiffness matrix.

Salisbury claimed that the torsional stiffness matrix was not a diagonal matrix. Salisbury's method failed for his torque-controlled system as the manipulator approached any kinematic singularity where zero joint torques would result.

Taking a closer look at the expression  $\mathbf{J}^T \cdot \mathbf{K} \cdot \mathbf{J}$ , we observe that it contains dimensional inconsistency, and thus is physically meaningless.

A more robust approach for stiffness control is explored in this experiment. The definition of the stiffness matrix in this work is determined by the type of joints of the

manipulator arm. When a prismatic joint is involved, a definition of linear stiffness is a reasonable choice. When a revolute joint is discussed, a definition of the torsional stiffness is used. Thus, a diagonal angular-stiffness matrix can be obtained for a robot with revolute joints, such as the PUMA robot.

In this experiment, the wrenches measured by the wrist-mounted force/torque sensor are transformed into joint torques as in equation (4.2). The resulting joint torques are divided by the specified torsional stiffness of each joint to obtain joint displacements. The torsional stiffness for each joint can be adjusted to a desired value by the software.

The results of the on-desk experiment showed that the robot responded perfectly -- application of a "fingertip" to the force/torque sensor produced a desired response on the part of the PUMA robot. Even when the operator shook his hand on the sensor, the PUMA robot duplicated the motion sensitively. Since there was no interaction between the operator's hand and the robot, force information after transformations served just as pure angular-displacement commands to the system with no dynamics involved. A prediction was made for those devices producing pure position commands to the system, so that similar "nice" responses could be expected.

Similarly, if the pure force information was fed from a remote manipulator to the controller, it could be used to generate a pure displacement command to the controller.

#### 4.2 Experiments on a Single-Axis System

The application of a force/torque sensor in a robotic joystick subsystem is different from its applications in the "fine motion" controls, such as deburring and assembly operations. In these applications, the robot is required only to move with small displacements and small control gains. On the other hand, the application of joystick tends to bring about gross motions in its whole working space with no pre-specified paths. Small control gains are thus inadequate.

Appropriate mathematical models help to develop intuitive understanding of system behavior. In the early stages of designing control strategies, a mathematical model of the robot system was attempted. Transfer functions of joints were identified experimentally. The complication of the complete model of the robot makes it impractical for real-time control. The combination of these single DOF linear models is, however, severely limited in its ability to present the real system, which includes nonlinearities. As the configuration of the robot changes, the behavior of the robot is made more complicated.

Pragmatically, control-gain limits were eventually established experimentally.

When the force/torque sensor is attached to the robot, dynamic interaction between the operator and the sensor is involved in the system operation. System stability suddenly becomes a critical issue in order for the PRJS to perform well.

In attempting to reduce the excitation of undesirable dynamic modes, several strategies were tested. These trial strategies were applied to the PRJS under an assumption that independent joint control of the PUMA controller appropriately takes care of the servo. Therefore, only a sufficiently smooth command profile needs to be generated for each joint. A threshold filter for each degree of freedom of the measured wrenches is set in order to filter out small vibrations from the operator's hand. The PRJS can be moved only if the applied wrenches are larger than the threshold level. The angular displacement commands are obtained by applying these strategies with the Jacobian matrix and the required coordinate transformations to the measured wrenches.

The trial strategies are as follows:

- 1)  $\theta = C_1 T,$
- 2)  $w = C_2 T,$
- 3)  $\alpha = C_3 T,$  and
- 4)  $w(i) = [(T(i-2) + T(i-1) + T(i))] \times C_4 / 3,$

where  $C_1$ ,  $C_2$ ,  $C_3$ , and  $C_4$  denote the control gains which are determined by experiments and  $T(i)$  represents the measured joint torque from the  $i$ -th control cycle. Angular velocity and angular acceleration are further used to generate angular displacements for commanding the PUMA robot.

The first trial strategy chooses joint torques as functions of angular displacements. In other words, stiffness constants are chosen so that joint torques can be converted into proportional angular displacements,  $\theta = C_1 T$ . Similar strategies have been used by Raibert and Craig and other researchers in hybrid control [26, 27]. Unlike the situation of the on-desk sensor, the experimental results show that the system is stable within a very narrow bandwidth, which agrees with indications given by Figure 3.5. Only when the stiffness constants are very large is the PRJS able to move smoothly with small displacements. When small stiffness constants are used, the joint oscillates and makes the controlled system unsuitable as a robotic joystick subsystem.

When joint torques are specified to be higher-order functions of angular displacements, less excitation of system dynamics are expected and thus the bandwidth of the dynamic ranges will be wider than that of the angular displacement function. The second trial strategy chooses joint torques to be the first-order function of angular displacements,  $w = C_2 T$ . Under this circumstance, the system can absorb more energy than the situation of the first trial strategy, and the PRJS

can be moved around by the operator with good speed. The only problem observed in this case is that the feeling of the PRJS in the operator's hand is a little heavy. Although different values of damping can be adopted to adjust the feeling, system stability limits the performance. Whenever small damping is used, less energy is dissipated and the PRJS feels lighter in the operator's hand. The boundary of stability can be determined experimentally.

The third trial strategy uses joint torques as a second-order function of angular displacements,  $\alpha = C_3T$ . With this strategy, the system is quite stable and the PRJS feels even lighter than in the previous two cases. A boundary of stability still exists. While the PRJS is made to feel lighter with this strategy, it is also easier for the system to drift.

In general, with higher-order functions, the PRJS has a wider stable range, feels lighter in the operator's hand, and its motion can easily be sped up. However, there is a drawback in using high-order functions for the PRJS. The trigger signal to the PRJS is the operator's application of force. With a first-order function, the PRJS will remain stationary when there is no force applied to its handgrip. With a second- or higher-order function, chances are that when the applied forces are removed from the PRJS's handgrip, it will still move. Under this situation an extra strategy is needed to stop the motion of the PRJS, and a time delay occurs

between the moment when the operator instructs the PRJS to stop and the moment when the PRJS stops its movement. A scary situation can take place when joint torques are specified to be second-order functions of joint displacement. The joint can be sped up with only a small amount of force applied. When joints are moving at high speed, it is hard for the operator to feel his application of wrenches to the grip. The operator may illusively believe he has no control of the joints. With the coordinated motion of multiple joints, the situation is made worse. Therefore, second- or higher-order functions are not recommended for application to a joystick. From this point of view, it seems the lower-order functions are natural candidates if they can be made more stable.

In order not to excite the system dynamic modes, a gentle transition of the velocity profile is necessary. The fourth trial strategy is to average the joint torque from three measurements,  $w(i) = [(T(i-2) + T(i-1) + T(i)) \times C_4]/3$ . By using this method, the system turns out to be more stable and lighter than the case of strategy two, and the performance can be tuned to be close to that of strategy three.

Except for strategy one, which has a very narrow stable range, the other strategies are further analyzed. In a series of experiments, the PRJS is requested to perform similar motions. The histories of joint torque and joint angle of joint 3 are recorded so that a comparison can be made. Initially, joint torques are zero before the motions take

place and joint 3 stays around 20 degrees with respect to its local coordinate system. When joint torques are applied, the PRJS is able to be moved as a joystick. The maneuverability of the PRJS and its sensitivity to the operator's instructions are important performance factors.

The recorded data are plotted for 14 seconds in Figures 4.2 to 4.4, respectively, with a different gain for each case. In general, based on the recorded data, a time lag between the commanded torque and the responding angular displacement exists; however, the operator's hand can barely sense this delay.

From these plots, one can see that the responding period for motion to take place is almost the same for all cases. This indicates that the time period which is occupied by the execution of force measurement and computations is more or less the same for all cases. What makes a difference is the time period in which the motion of the joint finishes the command.

The performance results for strategy two, strategy three, and strategy four are summarized as Table 4.1, regarding their response, maneuverability, lightness, drifting, and boundaries of stability. Among them, the performance of strategy four is between strategy two and strategy three. When the control gains are tuned to large values for strategy four, its performance can be almost as good as that of strategy three, with no drifting. In conclusion, strategy four is a good

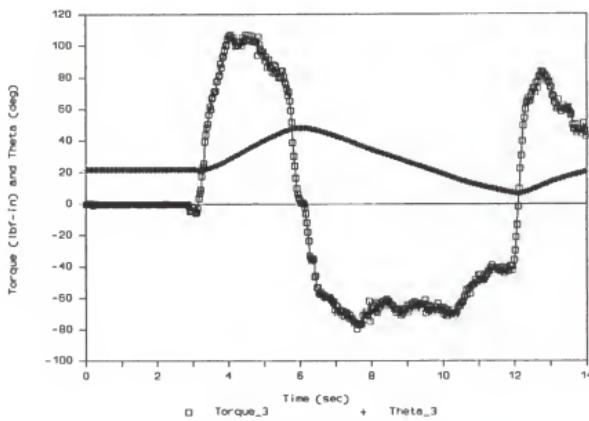


Figure 4.2(a)  $\Theta_3$  Response of Velocity Function with Gain = 0.5

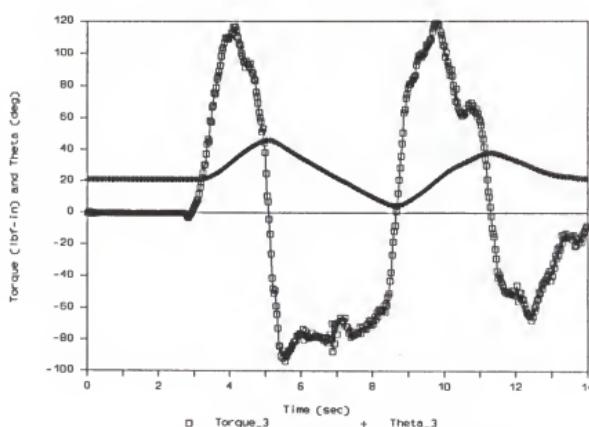


Figure 4.2(b)  $\Theta_3$  Response of Velocity Function with Gain = 0.7

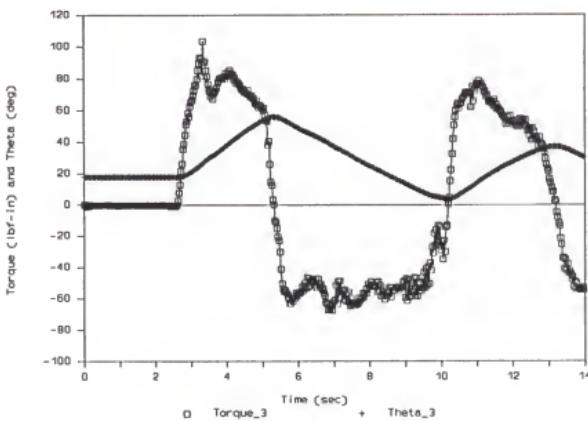


Figure 4.2(c)  $\theta_3$  Response of Velocity Function with Gain = 0.9

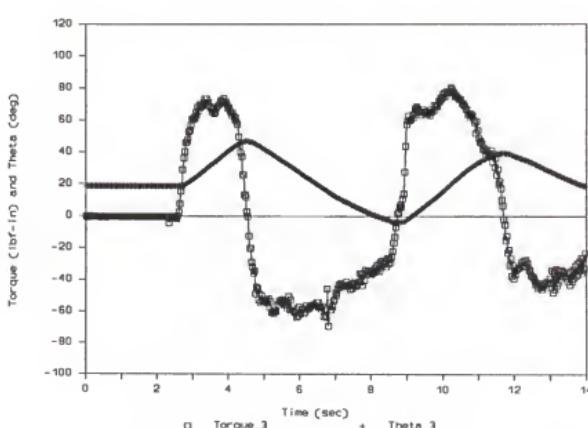


Figure 4.2(d)  $\theta_3$  Response of Velocity Function with Gain = 0.11

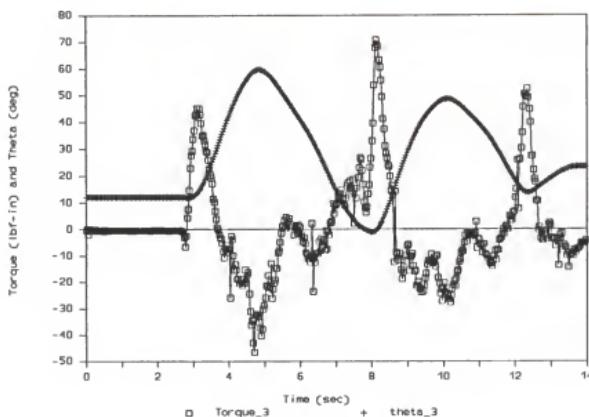


Figure 4.3(a)  $\theta_3$  Response of Acceleration Function with Gain = 0.2

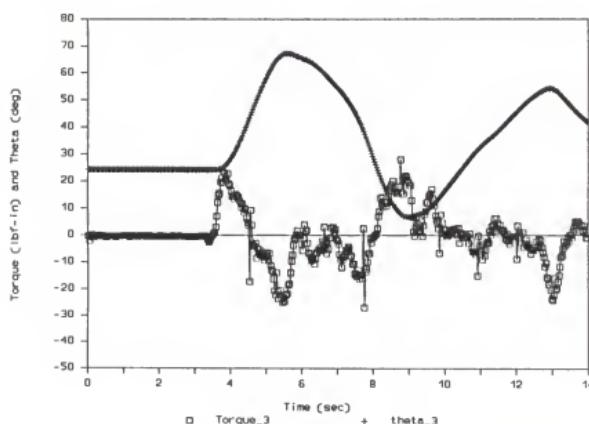


Figure 4.3(b)  $\theta_3$  Response of Acceleration Function with Gain = 0.3

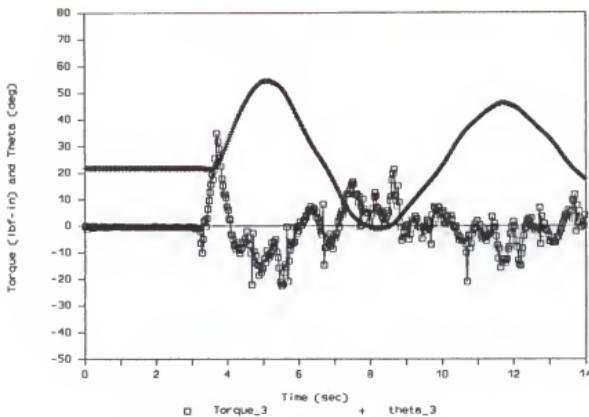


Figure 4.3(c)  $\theta_3$  Response of Acceleration Function with Gain = 0.4

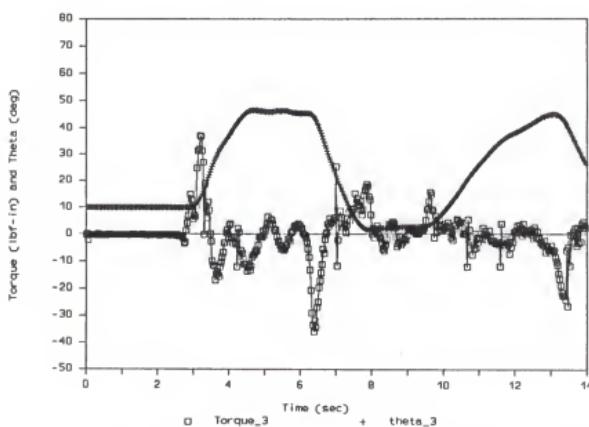


Figure 4.3(d)  $\theta_3$  Response of Acceleration Function with Gain = 0.5

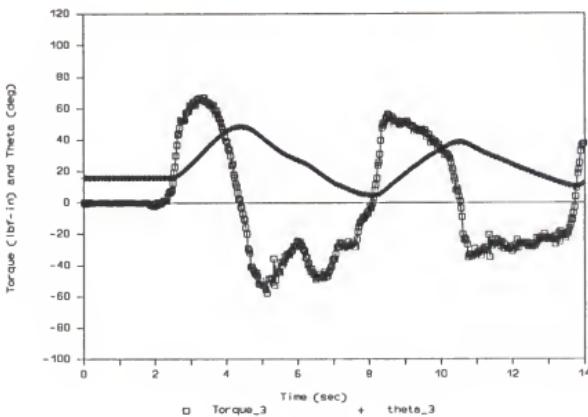


Figure 4.4(a)  $\theta_3$  Response of Average Velocity Function, Gain = 0.5

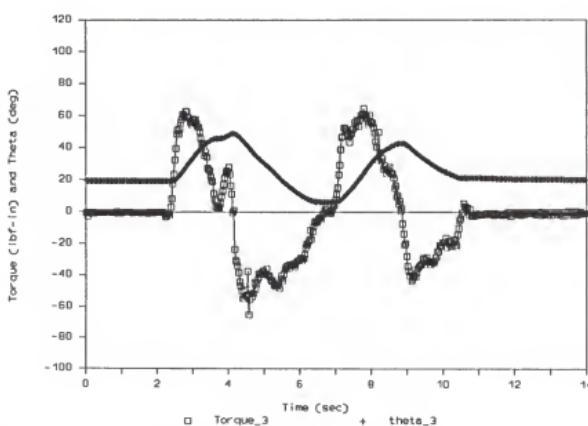


Figure 4.4(b)  $\theta_3$  Response of Average Velocity Function, Gain = 0.7

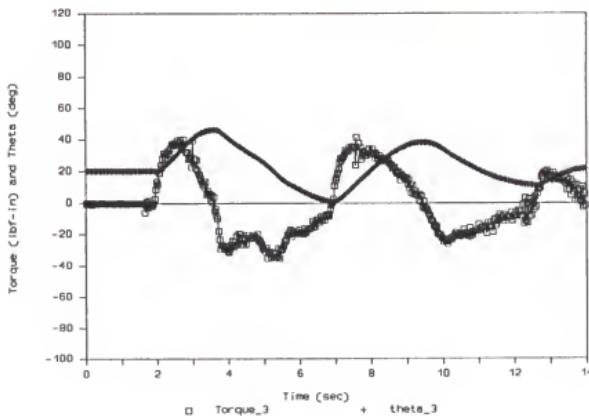


Figure 4.4(c)  $\theta_3$  Response of Average Velocity Function, Gain = 0.9

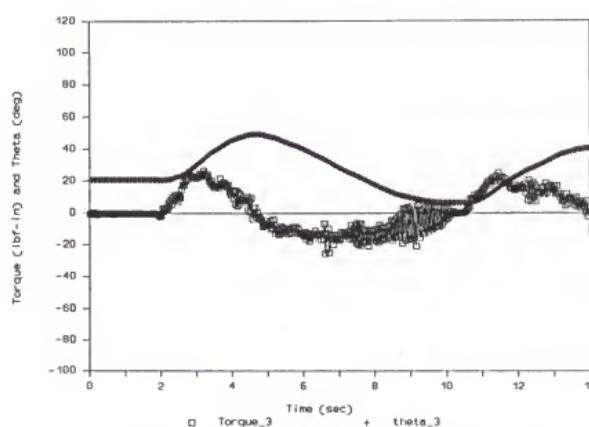


Figure 4.4(d)  $\theta_3$  Response of Average Velocity Function, Gain = 0.11

candidate for serving the purpose of a robotic joystick subsystem which provides a nice feeling on the operator's hand. It has been adopted in the remote force-reflection joystick experiment.

Table 4.1 Comparison of Performance

	<u>Strategy 2</u>	<u>Strategy 3</u>	<u>Strategy 4</u>
Response	Slowest	Fastest	Close to 3
Maneuverability	Worst	Best	Close to 3
Lightness	Heavy	Light	Moderate
Drifting	No	Yes	No
Stable Boundary	Low	High	Moderate

#### 4.3 Experimental Observations for the PRJS

A reason that robot manufacturers hesitate to integrate force/torque sensors into commercially available robots is the stability problem. When a wrist force/torque sensor is attached to the end of a serial-link robot, the bandwidth of the system is critically reduced. Thus, the system is easily driven into oscillations, even when a small load is applied. Dynamic coupling effects between links deteriorate its performance even more. To use the PUMA robot as a multiple-degree-of-freedom joystick by means of integrating a wrist

force/torque sensor, the adopted control gains need to be smaller than their marginal limits obtained from each single-joint experiment. The experimental results confirm this. When the marginal gains, which were measured from single-joint experiments, were used for multiple-joint motions, especially with non-planar motions, the PRJS oscillates severely. With lower gains, the PRJS became stable and easy to manage.

Another observed phenomenon is that the PRJS actually feels lighter than each single-joint system. This is because the PRJS with multiple-joint motion can respond to not only a single-degree-of-freedom torque, but also all the directions of wrenches. The PRJS is very sensitive to the operator's application of wrenches. Different operators, who have different effective damping, generate oscillations with different control gains. The PRJS becomes unstable at different control gains for an operator who uses stiff hand as opposed to one who uses a soft hand.

When limitations to the operating ranges of the PRJS are required, boundaries can be set in programming codes. Experiments have been performed with "hard boundaries," which apply inward constant torques to the assigned working ranges. When a joint is moved out of its working range, the inward torque is applied. If the joint is kept outside the assigned working range, the operator feels a constant torque pushing the joint back to its working range. If the operator releases the joint outside of its working range, the joint moves back

into its working range. When a large inward step torque is used for the PRJS response, a small oscillation can arise if the operator pushes a joint outside of its boundary using a "stiff" grip.

For each joint, when it is rapidly shaken, the operator feels servo frequency. This indeed exists for every force-reflecting joystick system. As long as this feeling is kept consistently small, it helps the operator to feel that the joystick is under control.

In conclusion, the PRJS with strategy four can be adjusted to operate smoothly at moderate speeds. As far as the application of joystick controller is concerned, the experiment shows promising results. The only flaw with the PRJS is that the handgrip is too heavy. The force/torque sensor senses wrenches when the handgrip is away from its initialization orientation. A lighter handgrip should be installed so that the weight of handgrip can be filtered out in the controlling program. Although the controlling program can theoretically be made to filter the effects of a heavy handgrip, a preferable solution is to design a substantially lighter handgrip.

#### 4.4 Remote Force-Reflection Joystick

To implement an experimental system with remote force reflection, the host computer and the remote computer are required to work tightly together. One has to notice that the

PUMA robot runs at a specified cycle time of 28 msec. In spite of all the other computations and data communications, the host computer must send joint commands every cycle to joint microprocessors in order to maintain the servo.

To incorporate the remotely reflecting force signal into the PRJS, the control program must be executed with minimum interference. The time period for data transmission must be made as short as possible. In the meantime, data transmission must not happen in the time period when servo operations are taking place.

With the existing facilities in CIMAR, another TANDY 4000 personal computer is used as a remote computer. Data communication between these two computers is through a serial line. In order to speed up the rate of data communication, the control chip of the serial port is programmed, and data are transmitted at 38400 baud.

Besides the communication speed, handshaking of data transmission is also important. Since the host computer for the PRJS needs to maintain the servos in real-time operation, it takes charge of the time when data transmission occurs in a control loop. The remote computer, once receiving a notice for data transmission, must respond promptly.

To take care of this situation, the procedures in the host computer and the remote computer are integrated with some degree of parallel-processing ability. The procedure in the remote computer is designed with an interrupt-driven function

by the input of the serial port. Thus, when it receives a notice for data transmission, the assigned data are transmitted. When there is no request for data transmission, the remote computer displays and updates the information according to keyboard instructions. With this asynchronous communication, the sending process of the remote computer does not have to suspend in order to wait for the acknowledgment signal from the receiving process of the host computer.

In this experiment, the remote computer can issue artificial torques to the robotic joystick subsystem. An artificial force in the Z-direction is applied to the system to demonstrate the force reflection. Its level can be increased with the "↑" key or decreased by the "↓" key. Other keys on the remote computer turn on/off the force reflection. When the operator grasps the handgrip of the PRJS, wrenches are reflected to the operator's hand. If the operator does not overcome them, the PRJS will push along corresponding directions with small displacements.

A more flexible program is also coded in the remote computer. It allows the operator to control the levels of reflecting torques for each of the six joints. The key commands are listed as followings:

- ↑ -- to increase the torque level,
- ↓ -- to decrease the torque level,
- -- move to modify the next joint torque,

← -- move to modify the previous joint torque, and  
other keys -- to turn on/off the force reflection.

The experimental system can reflect the remotely artificial torques. System response is stable and smooth.

#### 4.5 Milestones for Achieving Experimental Results

In robotics research, data communications and system integration have been ignored topics although they are necessary components to a complete system. Recently, Brady reviewed "robotics science" and recognized system integration as an open problem [56]. These topics are occasionally studied in theory; though a practical solution is still required. Ignoring these topics is often reflected by a degradation of system performance. To obtain good performance with integrated systems, the integration must be managed with an engineering concept of global optimization. All computations and data communications must be taken into consideration and must be the most effective configuration within the economical conditions. In order to have a grasp on every piece of a system, off-the-shelf softwares are not appropriate since they carry too much system overhead and "dark corners" which could trap end users.

In integrating the PRJS system, data communications, interface compatibility, and the requirement of real-time operation are significant problems encountered.

Reliable networking with high communication speed should be adopted for data communications if it is available. Parallel interface is superior to serial interface as far as communication speed is concerned. To determine the transmitted data between systems, a specific format regarding both data type and data block must be chosen. The thought of reserving the transmitted data space for later usage is often too generous to be tolerated. As long as a byte (8 bits) is sufficient to cover the range of data, a 2-byte data type is wasteful.

In the PRJS, instead of using a serial port of the PUMA controller, a parallel connection is made between the TANDY computer and the AIB of the PUMA controller. Another parallel interface is used to read the sensed wrenches from FT-15/50 so that not only the highest sampling rate of the force/torque sensor but also the shortest transmitting time of the wrench data is obtained. The communication software is coded specifically for these purposes and a tight communication is implemented between the host computer and the PUMA robot system. Limited by the existing facilities, a serial line is used to connect the host computer and the remote computer. The deficiency is partially overcome by programming the control chip (NS 16450) for the serial port inside the TANDY computer. The communication speed of 38600 baud is thus reached, which is four times higher than common serial communication speed. To demonstrate the feasibility of the

reflecting forces from a remote system to the PRJS, joint-torque information is transmitted from a remote computer to the host computer in a one-byte format for each DOF. An interrupt programming technique is applied to make the two computer systems work nearly independently. The arrangement of the minimum communication time makes this simulated bilateral system operate transparently.

Accessibility to the system hardware is an important feature for personal computers, which is normally not allowed with mainframe systems. The performance of integrated systems can thus be improved. The only hurdle with hardware-programming techniques is that they are system-dependent and most of these technique details are not mentioned in manuals. BIOS function calls and DOS function calls are listed in the technique manual without specific details. Extensive experimentation is required in order to take advantage of the computer system hardware. In controlling the PUMA robot, the computer is required to track the 28 msec cycle time. The address of system timers can be directly accessed, but they count only up to a power of two multiplied by 1.935125 milliseconds, which is not suitable for the application of the PRJS. The BIOS interrupt, 15H, with registers AH = 85H, AL = 0, DX = 1, can be used for this purpose. This function call offers the ability to count milliseconds as a background task. Therefore, no extra CPU load is added in executing the programming codes. A puzzle about the usage of this BIOS

timer function was encountered. It apparently could be triggered by some tasks in the computer system and never be turned off. Under such circumstances, the BIOS function did not work properly. This was remedied by resetting the timer in order to assure the BIOS call serves its function.

Another difficulty in the PRJS is the compatibility of interfaces. The parallel interface, PC-11, is the only known hardware which "emulates" the DRV-11 parallel interface for the PC-bus. It is important to realize that the 8-bit PC-bus PC-11 is designed to emulate the 16-bit Q-bus DRV-11 parallel interface. In order to assure the interfaces are compatible up to the functional needs, experiments are necessary to confirm the exact results. This is important in building the operator's confidence before grasping the endgrip of the PUMA robot as a joystick device. Experimental results showed that a dummy variable must be sent to the DROUTH register to ensure against a data overrun problem with the PC-11 interface. Unexpected motion of the PUMA robot may occur if the DROUTH register does not receive a variable every executing cycle.

Real-time operation requires all the computations to be finished in less than 28 milliseconds. The computation codes must be arranged without any redundant operations. The control algorithm must also be simple enough to implement. The existing facilities in the laboratory make it difficult for a real-time system to log data without disturbing system performance.

Another caution about feedback control is that the loop must be closed in a very tight fashion. Computations should be executed outside the feedback loop as much as possible. Otherwise, joints could travel some distances before the feedback signals are summed to the input commands and thus extra errors are created.

CHAPTER 5  
CONCLUSIONS AND RECOMMENDATIONS

5.1 Concluding Remarks

The experiment has successfully demonstrated the feasibility of utilizing an industrial robot as a joystick controller. This success is no miracle. By inspecting the technology of the design of robotic systems, we can recognize that the design of a joystick controller and that of a robot system share the same technological foundation. It is also important to notice that the approach in this project can be readily implemented on any existing industrial robot by enhancing the system with an extra force-detecting sensor, for example a force/torque sensor. Reliability and stability of the additional sensor are important.

In the case of robotic systems, compromises between important performance parameters such as payload and the maximum velocity are normally made. Payload for a joystick controller is not significant because it is undesirable to impress large forces on an operator (especially for long periods of time). Therefore, a wider performing range may be adopted in designing a joystick controller.

In attempting to use an existing robot system as a joystick, it is good to know the advantages and disadvantages associated with this idea. The slave manipulator system and communication systems are assumed to be modular so that the comparison can be made on the same basis. Under such circumstances, a robotic joystick subsystem contains merits such as:

- The design process is essentially no longer required.
- It can provide a testbed for different types of joysticks as long as those types of robots exist.
- There is no need to be bothered about the use of analog-to-digital converters in a robotic joystick subsystem since they are already built into the controller of the robot system.
- A robotic joystick system is more flexible in the sense that system parameters, such as damping, stiffness, and gain, are programmable. On the contrary, a designed mechanical device always comes with specific (though not necessarily well known) mechanical properties.

There are also some demerits that come along with this approach. Comparing with a designed joystick controller, a robotic joystick subsystem needs the computer system to carry, in some respects, an extra burden; for instance, the I/O to an interface for the force/torque sensor in this project is

not normally utilized in a robot system. Extra computations of the Jacobian matrix and its coordinate transformations are also not common in an industrial robotic system.

To use an industrial robot as a joystick, there are some issues with no clear-cut answer to say whether they are good or not. For example, when the servo of the factory-supplied system is adopted, no work is required in order to tune the servo gains. The responses of the servos are, however, limited by the factory design.

Besides all these comparisons, the PRJS does show the capability of being used as a joystick regardless of its physical dimensions.

### 5.2 Recommendations and Future Work

The main goal of this project is not to rule out the contribution of research on the design of teleoperation controllers. We hope to make more effective and more profitable use of the existing knowledge and technologies.

One may wonder who will pay for a joystick controller at the cost of a robot system. However, joystick controllers for the bilateral system do carry the same components as the robots. The expense of the bilateral controller can be reduced only if coarse resolution potentiometers are used, instead of the fine resolution encoders. Besides, the robot manufacturers must recognize the diverse applications of their technologies.

Based on the experiences of this project, it is believed that several actions can be taken to improve the performance of a robotic joystick subsystem.

An overall review of the existing industrial robots is helpful in that a better candidate for a specific purpose can be made. The operating volume of the robot is one thing to be concerned about. A good joystick should comply with the workspace of a human being. The sampling rate is another crucial factor for the robotic joystick subsystem. A faster sampling rate can make the system work closer to a continuous-time system, and time delay in the response of joint motions can be reduced. This concept can be analogously regarded as a mechanical system with less backlash.

It is obvious that there is more than one method to convert the industrial robot to a joystick, depending upon the available facilities. As far as the enhanced sensor is concerned, force/torque sensors can be alternatively put on the joint axis (the so-called colocated sensor) so that a larger bandwidth of a stable dynamic range is obtained and bigger control gains can be used. If a suitable non-attached sensor to the robot is integrated into the system, none of the stability of dynamics response will have been affected.

An appropriate design of the computer architecture can make the whole system perform well. A tightly coupled distributed computer system is mandatory for a good teleoperator system. Although the concept of networking

offers high-speed data communication, experience did not show it to be beneficial all the time. It is better if a shared-memory architecture exists in the system so that no delay of data distribution takes place between processes. An operating system with multitasking capability can be helpful in robot control if system overhead can be avoided.

Safety has been an important consideration by the users of robots. In a robotic joystick subsystem, suitable safety measurements must be adopted. This is especially true for a force-reflecting joystick.

An additional benefit of this project is to open the view of the robot manufacturers. Owing to the similarity between the robot and the joystick, the manufacturers could extend their own technologies to joystick design and production.

## REFERENCES

1. Johnsen, E. G., "Telesensors, Teleoperators, and Telecontrols for Remote Operations," IEEE Trans. on Nuclear Science, Vol. NS13, 1966, p. 14.
2. Goertz, R. C., "Manipulators Used for Handling Radioactive Materials," Human Factors in Technology, Chap. 27, ed. Bennett, E., McGraw-Hill, New York, 1963.
3. Tesar, D., and Lipkin, H., "Assessment for the Man-Machine Interface Between the Human Operator and the Robotic Manipulator," NSF Grant ENG78-20112 and DOE contract ER-78-S-05-6102, University of Florida, 1978.
4. Schultz, R. C., "Computer Augmented Manual Control of Remote Manipulators," Master's Thesis, University of Florida, 1978.
5. Lipkin H., "Kinematic Control of a Robotic Manipulator with a Unilateral Manual Controller," Master's Thesis, University of Florida, 1983.
6. Tesar, D., Tosunoglu, S., and Lindemann, R., "Construction and Demonstration of a 9-String 6-DOF Force Reflecting Joystick for Telemanipulation," University of Texas, Austin, Oct. 1987.
7. Brown, T. D., "The Design and Fabrication of a Four Degree-of-Freedom Force-Feedback Planar Manual Controller," Master's Thesis, University of Florida, 1984.
8. Walker, C., "The Control and Implementation of a Four Degree-of-Freedom Force-Feedback Planar Manual Controller," Master's Thesis, University of Florida, 1984.
9. Wegerif, D. G., "Control of Redundant Degree-of-Freedom Planar Force-Feedback Joystick," Master's Thesis, University of Florida, 1984.
10. Morton, A. T., "Design and Implementation of a Joystick Based Teaching System," Ph.D. Dissertation, University of Missouri-Rolla, 1985.

11. Bejczy, A. K., and Salisbury, J. K., "Controlling Remote Manipulators Through kinesthetic Coupling," Computer in Mechanical Engineering, Vol. 2, July 1983, pp. 48-60.
12. Hirzinger, G., "Robot-Teaching Via Force-Torque Sensors," European Meeting on Cybernetics and Systems Research, Austria, 1982, pp. 955-963.
13. Hirzinger, G., "Robot Learning and Teach-In Based on Sensory Feedback," Robotics Research: The Third International Symposium, The MIT Press, Cambridge, Massachusetts, 1986, pp. 155-163.
14. Hirzinger, G., "Sensory Feedback in the External Loop," Proc. IFAC on Robot Control (SYROCO '85), Spain, 1985, pp. 445-451.
15. Hirzinger, G., "The Space Teleoperatic Concepts of DFVLR Rotex," Proc. IEEE Int'l Conf. on Robotics and Automation, 1987, pp. 443-449.
16. Hirzinger, G., "Sensory Feedback in Robotics - State-of-the-Art in Research and Industry," Proc. Tenth Triennial World Congress of IFAC, Vol. IV, Munich, FRG, 1987, pp. 193-206.
17. Whitney, D. E., "The Mathematics of Coordinated Control of Prosthetic Arms and Manipulators," Trans. of the ASME Journal of Dynamic Systems, Measurement, and Control, Vol. 94, Dec. 1972, pp. 303-309.
18. Wilt, D. R., Pieper, D. L., Frank, A. S., and Glenn, G. G., "An Evaluation of Control Modes in High Gain Manipulator Systems," IFToMM Mech. & Mech. Theory, Vol. 102, 1977, pp. 373-386.
19. Brook, T. L., "Superman: A System for Supervisory Manipulator and the Study of Human/Computer Interaction," Master's Thesis, MIT, 1979.
20. Tesar, D., and Butler, M. S., "A Generalized Modular Architecture for Robot Structure," ASME Manufacturing Review, Vol. 2, No. 2, June 1989, pp. 91-118.
21. Vertut, J., and Coiffet, P., Teleoperations and Robotics: Evolution and Development, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1986.
22. Brooks, T. L., and Bejczy, A. K., "Hand Controllers for Teleoperation: A State-of-the-Art Technology Survey and Evaluation," JPL pub. 85-11.

23. Shieh, C., "The Control of PUMA 600 Robot Without Using VAL," Master's Thesis, University of Florida, 1986.
24. Unimation PUMA Robot Manual, Text 398H1A, Vol. 1, Unimation Inc., Danbury, Connecticut, April 1980.
25. Inoue, H., "Computer Controlled Bilateral Manipulator," Bulletin of the Japan Society of Mechanical Engineer, Vol. 14, March 1971, pp. 199-207.
26. Mason, T. M., "Compliance and Force Control for Computer-controlled Manipulators," IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-11, No. 6, June 1981, pp. 418-432.
27. Raibert, M. H., and Craig, J. J., "Hybrid Position/Force Control of Manipulators," Trans. of the ASME Journal of Dynamics Systems, Measurements, and Control, Vol. 103, No. 2, June 1981, pp. 126-133.
28. Hollerback, J. M., "A recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity," IEEE Trans. on Systems, Man, and Cybernetics, SMC-10, Nov. 1980, pp. 730-736.
29. Luh, J. Y. S., Walker, M. W., and Paul, R. P. C., "On-line Computational Scheme for Mechanical Manipulators," Trans. of the ASME Journal of Dynamic Systems, Measurement, and Control, Vol. 102, June 1980, pp. 69-76.
30. Khatib O. and Burdick J., "Force Control of Robot Manipulators," Proc. of the 7th World Congress of the Theory of Machines and Mechanisms, Spain, Sept. 17-22, 1987, pp. 1213-1218.
31. Whitney, D. E., "Force Feedback Control of Manipulator Fine Motions," Trans. of the ASME Journal of Dynamics Systems, Measurements, and Control, Vol. 99, June 1977, pp. 91-97.
32. Salisbury, J. K., "Active Stiffness Control of a Manipulator in Cartesian Coordinates," Proc. 19th IEEE Conf. on Decision and Control, Dec. 1980, pp. 95-100.
33. Whitney, D. E., "Historical Perspective and State of the Art in Robot Force Control," The International Journal of Robotics Research, Vol. 6, No. 1, Spring 1987, pp. 3-14.
34. Hogan, N., "Impedance Control: An Approach to Manipulation: Part I - Theory, Part II - Implementation, Part III - Applications," Trans. of the ASME Journal of Dynamic Systems, Measurement and Control, Vol. 107, pp. 1-24.

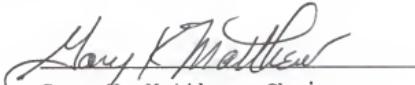
35. Simons, J., and Van Brussel, H., "Force Control Schemes for Robot Assembly," Robotic Assembly, ed., Rathmiller, K., IFS Publications, Berlin, 1985, pp. 253-265.
36. Maples, J. A., and Becker, J. J., "Experiments in Force Control of Robotic Manipulators," Proc. IEEE Int'l Conf. on Robotics and Automation, 1986, pp. 695-702.
37. An, C. H., and Hollerbach, J. M., "Kinematic Stability Issues in Force Control of Manipulators," Proc. IEEE Int'l Conf. on Robotics and Automation, 1987, pp. 897-903.
38. An, C. H., and Hollerbach, J. M., "Dynamic Stability Issues in Force Control of Manipulators," Proc. IEEE Int'l Conf. on Robotics and Automation, 1987, pp. 890-896.
39. Eppinger, S. D., and Seering, W. P., "On Dynamic Models of Robot Force Control", Proc. IEEE Int'l Conf. on Robotics and Automation, 1986, pp. 29-34.
40. Eppinger, S. D., and Seering, W. P., "Understanding Bandwidth Limitations in Robot Force Control", Proc. IEEE Int'l Conf. on Robotics and Automation, 1987, pp. 904-909.
41. Eppinger, S. D., and Seering, W. P., "Modeling Robot Flexibility for Endpoint Force Control", Proc. IEEE Int'l Conf. on Robotics and Automation, 1988, pp. 165-170.
42. Kazerooni, H., Houpt, P. K., and Sheridan, T. B., "The Fundamental Concept of Robust Compliant Motion for Robot Manipulators," Proc. IEEE Int'l Conf. on Robotics and Automation, 1986, pp. 418-427.
43. Lipkin, H., and Duffy, J., "The Elliptic Polarity of Screws," Trans. of the ASME Journal Mechanisms, Transmissions, and Automation in Design, Vol. 107, June 1985, pp. 377-387.
44. Lipkin, H., and Duffy, J., "Hybrid Twist and Wrench Control of a Robotic Manipulator," Trans. of the ASME J. Mechanisms, Transmissions, and Automation in Design, Vol. 110, June 1988, pp. 138-144.
45. Duffy, J., "The Fallacy of Modern Hybrid Control Theory that is Based on 'Orthogonal Complements' of Twist and Wrench Spaces," Journal of Robotics Systems, Vol. 7(2), 1990, pp. 139-144.
46. Installation and Operations Manual for F/T series Force/torque Sensing Systems, Lord Corporation, Cary, NC, 1986.

47. PC-11 User Guide, Omnicomp Graphics Corporation, Houston, Texas, Oct. 1987.
48. Orin, D. E., and Schrader, W. W., "Efficient Computation of the Jacobian for Robot Manipulators," The International Journal of Robotic Research, Vol. 3, No. 4, Winter 1984, pp. 66-75.
49. Waldron, K. J., Wang, S. L., and Bolin, S. J., "A study of the Jacobian Matrix of Serial Manipulators," Trans. of the ASME Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 107, June 1985, pp. 230-238.
50. Lipkin, H., and Duffy, J., "A Vector Analysis of Robot Manipulators," Recent Advances in Robotics, ed., Beni, G., and Hackwood, S., John Wiley & Sons, Inc., New York, 1985, pp. 175-241.
51. Hunt, K. H., "Robot Kinematics- A Compact Analytic Inverse Solution for Velocities," Trans of the ASME Journal of Mechanisms, Transmissions and Automation in Design, Vol. 109, March 1987, pp.42-49.
52. Soylu, R., "Kinematics, Special Configurations and Performance Evalution of Serial Manipulators," Ph.D. Dissertation, University of Florida, 1988.
53. Ball, R. S., A Treatise on the Theory of Screws, Cambridge University Press, London, 1900.
54. Lipkin, H., and Duffy, J., "Analysis of Industrial Robots via the Theory of Screws," Proc. of the 12th International Symposium on Industrial Robots, Paris, 1982, pp. 359-370.
55. Griffis, M., "Kinetic Considerations in the Hybrid Control of Robotic Manipulators," Master's Thesis, University of Florida, 1988.
56. Brady, M., "Introduction: The Problems of Robotics," Robotics Science, ed., Brady, M., The MIT press, Cambridge, Massachusetts, 1989, pp. 1-35.

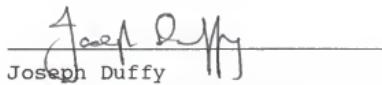
#### BIOGRAPHICAL SKETCH

Ching-Shyong Shieh was born in Hualian, Taiwan, R.O.C., on January 15, 1957. He was awarded the Bachelor of Science degree in mechanical engineering from the Tatung Institute of Technology, Taiwan, in 1980. After serving his military duty for two years, he worked as a research/teaching assistant at the department of mechanical engineering, National Sun Yat-Sen University during 1982 and 1984. He went to the University of Florida in September 1984 to continue his graduate studies. In December 1986, he was awarded the Master of Science degree in mechanical engineering. That year, he continued his graduate studies at the University of Florida.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

  
\_\_\_\_\_  
Gary K. Matthew, Chairman  
Associate Professor of  
Mechanical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

  
\_\_\_\_\_  
Joseph Duffy  
Graduate Research Professor of  
Mechanical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

  
\_\_\_\_\_  
Keith L. Doty  
Professor of  
Electrical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

  
\_\_\_\_\_  
Carl D. Crane, III  
Assistant Professor of  
Mechanical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

George N. Sandor

George N. Sandor  
Research Professor of  
Mechanical Engineering

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

December 1990

Hubert C. Bewe  
for Winfred M. Phillips  
Dean, College of Engineering

---

Madelyn M. Lockhart  
Dean, Graduate School